

nº 12 160 ptas (IVA incluido)

M.U.L.E.: UN SIMPATICO MONOPOLY INTERGALACTICO

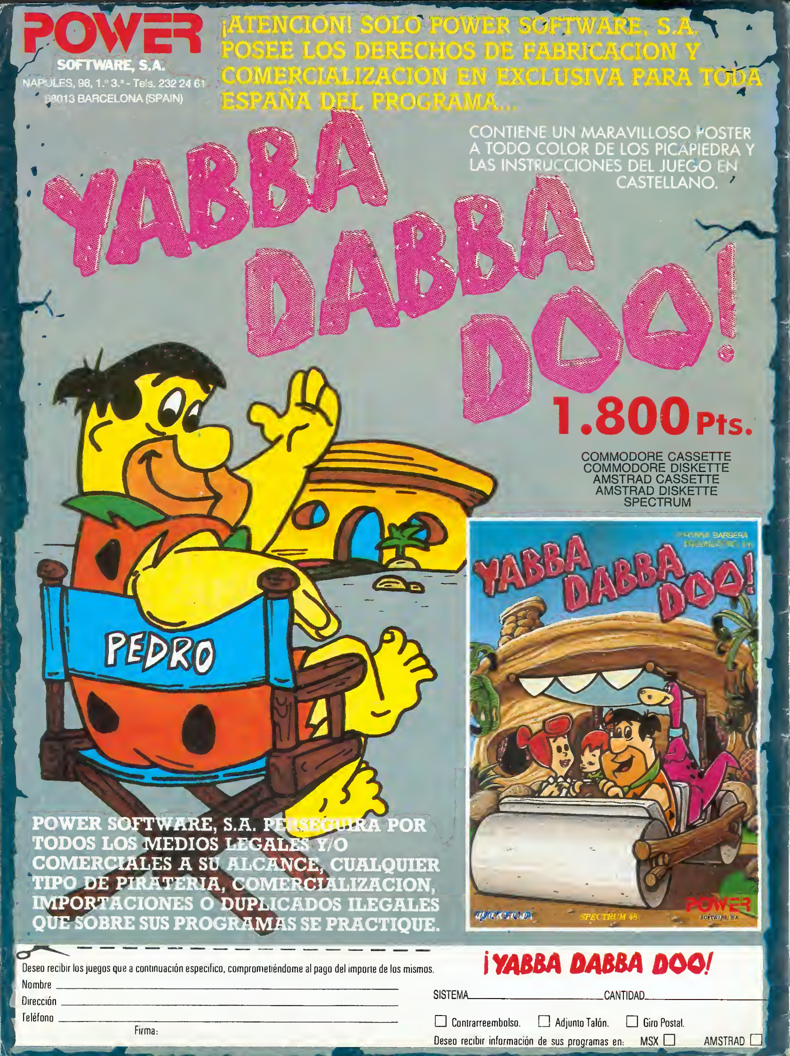
SEMANAL

perera.

VELOCIDAD COMPATIBLE PARA TODOS LOS COMMODORE

GUILLERMO TELL DE SALON

ESTRUCTURA DEL MICROPROCESADOR



POWER
SOFTWARE, S.A.
NAPOLES, 98, 1.º 3.ª - Tels. 232 24 61
08013 BARCELONA (SPAIN)
¡ATENCION! SOLO POWER SOFTWARE, S.A. POSEE LOS DERECHOS DE FABRICACION Y COMERCIALIZACION EN EXCLUSIVA PARA TODA ESPAÑA DEL PROGRAMA...
CONTIENE UN MARAVILLOSO POSTER A TODO COLOR DE LOS PICAPIEDRA Y LAS INSTRUCCIONES DEL JUEGO EN CASTELLANO.
YABBA DABBA DOO!
1.800 Pts.
COMMODORE CASSETTE
COMMODORE DISKETTE
AMSTRAD CASSETTE
AMSTRAD DISKETTE
SPECTRUM
PEDRO
YABBA DABBA DOO!
SPECTRUM 48
POWER SOFTWARE S/A
POWER SOFTWARE, S.A. PERSEGUIRA POR TODOS LOS MEDIOS LEGALES Y/O COMERCIALES A SU ALCANCE, CUALQUIER TIPO DE PIRATERIA, COMERCIALIZACION, IMPORTACIONES O DUPLICADOS ILEGALES QUE SOBRE SUS PROGRAMAS SE PRACTIQUE.
Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.
Nombre
Dirección
Teléfono
Firma:
¡YABBA DABBA DOO!
SISTEMA
CANTIDAD
Contrarreembolso.
Adjunto Talón.
Giro Postal.
Deseo recibir información de sus programas en: MSX
AMSTRAD



Director:
ANTONIO M. FERRER ABELLO

Redactor-jefe:
FERNANDO LOPEZ MARTINEZ

Redacción:
ANTONIO CARVAJAL
JUAN M. LOPEZ MARTINEZ
PABLO GARCIA MOLINA
JOSE LUIS DE DIEGO
IGNACIO BARCO LUENGO
ALFREDO SINDIN VALERO

Colaboradores
JOSE LUIS VAZQUEZ DE PRAGA
ANTONIO MANZANERA

Secretaría de Redacción:
PILAR MANZANERA AMARO

Diseño y Maquetación:
CARLOS GONZALEZ AMEZUA
MAITE CONDE

Ilustraciones:
ANTONIO PERERA
RAMON POLO

Fotografía:
Equipo Gálata

Directora Publicidad
CARMINA FERRER
Tel. 457 69 23

Publicidad Barcelona:
ISIDRO IGLESIAS
Avda. Corts Catalanes, 1010
Tel. (93) 307 11 13

Director de Producción:
VICENTE ROBLES

Directora de Administración:
MARIA ANTONIA BUITRAGO

Suscripciones:
MARIA GONZALEZ AMEZUA

Redacción, administración, publicidad y suscripciones:
Plza. República Ecuador, 2
28016 MADRID. Tel. 250 58 20
Télex 49371 ELOC E

Dirección para correspondencia:
Aptdo. de Correos 61.294
28080 MADRID

Fotocomposición:
VIERNA, S. A. MADRID

Fotomecánica:
RODACOLOR, S. A. MADRID

Imprenta:
GRAFICAS REUNIDAS, S. A. MADRID

Distribución:
COEDIS. Valencia, 245.
BARCELONA

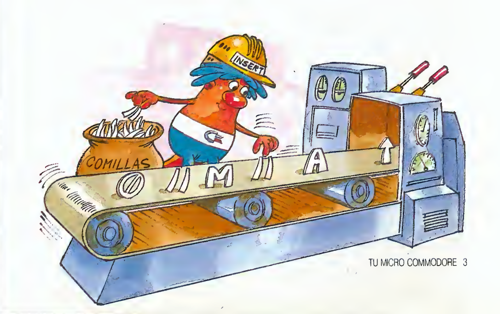
Precios para España. Ejemplar: (160 ptas. (IVA incluido). 155 ptas. (Canarias, Ceuta y Melilla). (La suscripción anual incluye 52 números.)

Distribución Cono Sur:
CADE, S.R.L.
Pasaje Sud América, 1532
Tel. 21 24 64.
Buenos Aires 1.290. Argentina.

Depósito Legal: M. 40920-1985.
Impreso en España.

SUMARIO

En esta semana, la sección de software nos aporta una serie de ideas muy a tener en cuenta para evitar la ralentización excesiva del, de por sí lento, lenguaje BASIC. Podremos observar mediante un programa repleto de ejemplos, los efectos que surten estos consejos, y lo que es más importante, podremos ser testigos de ellos todos: sí, tanto los que tengamos un C-64, como un C-16 o un C-128. Además, pequeño susto de la página de NOTICIAS, donde encontraremos una breve reseña sobre la recuperación de la crisis padecida a finales de año por Commodore International, que en absoluto afecta a su distribuidor: Microelectrónica y Control.



PEQUEÑOS PROBLEMAS ECONOMICOS EN COMMODORE: NO EXISTE NINGUN MOTIVO PARA ALARMARSE

El 31 de Enero pasado la multinacional norteamericana Commodore International anunció que sus acreedores le habían concedido un nuevo aplazamiento en la negociación de su deuda, encontrándose la empresa técnicamente en suspensión de pagos.

Este aplazamiento es sin duda debido a los síntomas de recuperación que presenta la compañía, que se encuentra atravesando un mal momento debido a la mala acogida (?!) que en el mercado norteamericano ha tenido el esperado Amiga.

Según datos procedentes de fuentes bien informadas, el fracaso inicial en el lanzamiento de este potente ordenador personal, ha producido unas pérdidas que rondan los 39,2 millones de dólares (28.800 millones de pesetas).

Dado que las conversaciones con los bancos continúan satisfactoriamente, ante la noticia de la reducción en el endeudamiento, Marshall Smith, presidente de la compañía, ha podido afirmar rotundamente que los rumores de quiebra están completamente desprovistos de fundamento.

Aunque debido a los misterios del marketing y la sociedad de consumo, el recibimiento dispensado al Amiga en su país natal ha dado un buen susto a sus creadores, no debemos olvidar la enorme difusión que los ordenadores Commodore tienen en todo el mundo, y seguramente el público de otros países haga una justicia mayor a este aparato.

Ya sólo resta desear a la compañía que continúe su franca recuperación, y supere el pequeño bache en que ha ido a tropezar.



EL SOFTWARE HA EMPRENDIDO SU LOCA E IMPARABLE CARRERA HACIA LA CUMBRE

Como todos sabéis, la palabra software se refiere a los programas y el uso de los ordenadores, puesto que los complejos lenguajes máquina no serían accesible más que para un reducido grupo de expertos y por tanto estos serían los únicos capaces de hacer funcionar a los ordenadores. Para que todos podamos manejarlos se han creado toda una serie de lenguajes y de programas.



Hoy en día hasta en el campo de la educación se han introducido los ordenadores, y ello es posible gracias a unos programas más accesibles. La creación y desarrollo de estos programas ha dado origen a una auténtica industria, en todo el mundo, cuyo crecimiento se estima según los expertos en el entorno del 30 al 50 por cien cada año.

Este hecho nos favorece mucho a todos, pues las utilidades siempre son necesarias y la concurrencia en el mercado de infinidad de juegos favorece su calidad e imaginación.

chispas

CHISPA

Por si no lo sabías, las segundas comillas de cualquier texto, dentro de una línea BASIC, no son necesarias (siempre que con ellas se finalizara esa línea), lo cual puede venir bien si no nos cabe la línea por un carácter. Así, tiene el mismo efecto **10 PRINT "HOLA"** que **10 PRINT "HOLA** (OFF).

CHISPA

Hay algunas sentencias de BASIC, que por no ser bien conocidas, o por que nadie las usa muy a menudo, se dejan en el olvido, despreciando una parte de la potencialidad de nuestro ordenador. En este caso, nos estamos refiriendo a **DEFine FuNction** (**DEF FN**). Por ejemplo, veamos como se puede aprovechar esta función definible, para efectuar un **POKE** en la memoria de pantalla, utilizando las variables X e Y.

```
10 DEF FNP(X)=1024
X+40*Y:PRINTCHR$
(147):C=54272
20 INPUT"X,Y";X,Y
:IFX<0ORX>39OR
Y<0ORY>24THEN15
30 POKE FNP (X),1:
POKE FNP(X)+C,1:FORI=0
TO999:NEXT:RUN,
```

Este sencillo programa, sitúa una «A» en blanco, en la posición de pantalla X (0-39) e (0-24): gracias a la función P(X), definida en la línea 10. Las X pasan de uno a uno, y las Y de 40 en 40; todo ello se suma a 1024 (inicio de la pantalla).

CHISPA

Cuando el ordenador ejecuta un IF, debe saber si la condición especificada se ha cumplido o no, para ello se generan dos valores; uno para verdadero (−1) y otro para falso (0), estos dos valores se pueden usar como si fuesen variables normales y corrientes, encerrando la comparación entre paréntesis. Para comprender mejor esto, veamos una aplicación que convierte una variable en un «interruptor»; cada vez que se defina valdrá 1, 0, 1, 0, etc.: **M=−(M 1).**

Si M=0, será falso que (M<>1) y esta expresión valdrá (−1); luego M = −(−1) → M=0.

Si M=1, será falso que (M<>1) y esta expresión valdrá (0); luego M = −(0) → M = 0.

CHISPA

Si queremos pasar una variable numérica positiva a cadena (por ejemplo para imprimirla eliminando espacios), no es suficiente ejecutar **A$=STR$(A),** puesto que esto genera un espacio delante de la cadena; este espacio es el correspondiente al signo, siendo un blanco para positivo o un guión (−) para negativo; para eliminarlo deberemos usar: **A$= MID$(STR$ (A),2).** Esto toma todos los caracteres a partir del segundo inclusive, eliminando así el indeseado espacio del signo.

CHISPA

Algunas veces es necesario escribir comillas dentro de un **PRINT,** pero como bien sabemos, no podemos emplear las accesibles desde el teclado con (SHIFT+2); en su lugar deberemos utilizar **CHR$(34),** veamos como:

Supongamos que queremos escribir el mensaje: **MI "TELE" ES BLANCA.** Para hacerlo usaremos el siguiente **PRINT:**

PRINT "MI "CHR$(34) "TELE"CHR$(34)"ES BLANCA"

CHISPA

De todos es sabido que en el C-64 no pueden coexistir mayúsculas y minúsculas, nada más que en los mensajes de pantalla (texto entre comillas); si queremos fijar un modo u otro, imprimir estos caracteres nos será de utilidad, para hacerlo podemos utilizar: **PRINT CHR$ (N);** donde N significa:

14: Cambia a presentación en minúsculas.

142: Cambia a presentación en mayúsculas.

8: Deshabilita el cambio mayúsculas-minúsculas con las teclas (COMMODORE +SHIFT).

9: Habilita el cambio mayúsculas-minúsculas con las teclas (COMMODORE + SHIFT).

CHISPA

Para los propietarios de un DÁTASSETTE, cargar un programa con autoejecución es fácil: pulsando las teclas SHIFT + RUN/STOP. He aquí el equivalente autorun en una unidad de disco: LOAD "nombre del programa",8: (aquí pulsar SHIFT + RUN/STOP).

CHISPA

Para simular un RESET en vez de apagar el ordenador y volver a encenderlo, se puede efectuar una llamada a la subrutina 64738, con SYS 64738 (¡ojo! se pierde el programa de la memoria); esta es la rutina que el sistema operativo accede al encender el ordenador.

CHISPA

Un buen método para salir del a veces fastidioso modo comillas del C-64 es pulsar a la vez las teclas SHIFT + RETURN, y regresar a la línea que se quiere modificar o introducir. SHIFT + RETURN efectúa un retorno de carro con alimentación de línea, pero sin ejecutar ni introducir la línea en memoria.

M.U.L.E.

Un «monopoly» del espacio donde cuatro jugadores tratan de sobrevivir en un mundo hostil y saturado de pinceladas de humor, con la única «ayuda» de las M.U.L.A. (Artilugios de Labores y Usos Múltiples).

La historia de las M.U.L.A. se remonta al momento en el que un importante constructor de maquinaria bélica comenzó su fabricación en un laboratorio subterráneo del Noroeste del Pacífico. Hasta aquí todo marchaba según el proyecto, pero la huelga de los robots de las cadenas de montaje forzó la intervención de una firma de suministros para restaurantes, un fabricante de muebles para preescolares y un magnate de las tostadoras, quienes acabaron el trabajo de una forma un tanto peculiar.

El resultado no fue el esperado, y difería en mucho del modelo inicial, lo cual repercutió en una serie de habilidades suplementarias propias de las M.U.L.A., tales como el don de captar cualquier transmisión intergaláctica, sintonización de juegos de pelota, cocina sencilla, frecuentes ataques de locura y un carácter rebelde que les incita a abandonar su trabajo en cuanto pueden.

A pesar de todos los inconvenientes asociados a estos ingenios mecánicos, su utilización para la conquista espacial es indispensable, pues son los únicos aptos para desarrollar este tipo de misiones. En el planeta Irata, escenario del juego, como en cualquier otro, cada colono debe mantener cortas las riendas de su M.U.L.A. y confiar en su apoyo a la hora de explotar los recursos naturales de dicho mundo.

LA SUPERVIVENCIA Y LA RIQUEZA SON LAS DOS MOTIVACIONES DE TODO COLONIZADOR

Una nave de exploración se ha posado en la superficie del planeta Irata, transportando a cuatro colonizadores con el objeto de permanecer vivos y obtener materias primas aptas. Al cabo de seis (nivel NOVATO) o doce meses (niveles NORMAL y de TORNEO), finalizará el plazo concedido a los viajeros espaciales, con el regreso de la nave y una valoración objetiva acerca de los logros conseguidos.

Ahora bien, una colonia subsiste si todos sus habitantes permanecen vivos y se desenvuelven con mayor o menor éxito. Un jugador es reconocido por el Commodore 64 como «Gran Fundador», es decir, el ganador indiscutible de la odisea, en caso de quedar primero en el sumario final y siempre que la riqueza total del planeta sea un valor concreto, el cual es indeterminado para un NOVATO, pero debe superar los 60.000 $ en los restantes niveles.

El dinero, medido en dólares ($), se obtiene directamente de la explotación del terreno y de su valor intrínseco (500 $), aunque un buen ejercicio de la compra-venta multiplicará rápidamente el capital. Por lo tanto, para triunfar en las finanzas galácticas no basta con cantar «Si yo fuera rico...», es preciso actuar de acuerdo con lo expuesto.

LAS OCHO RAZAS DEL UNIVERSO, POSIBLES PROTAGONISTAS DE LA CONQUISTA ESPACIAL

Una vez cargado el programa, la pantalla de presentación aparecerá ante nuestros sorprendidos órganos visuales (ojos), y nuestros instrumentos auditivos (oídos) percibirán los compases propios de la música del juego. Para acelerar este proceso considerablemente, nada mejor que oprimir el botón de disparo del joystick.

A continuación, sobreviene la elección del nivel de juego y del número de humanos participantes en el mismo. Este número varía desde cuatro hasta cero, correspondiendo la última modalidad al juego de demostración. Habiendo escogido ya, los jugadores presionarán de nuevo el disparador y/o pulsarán la tecla correspondiente, accediendo así a la elección de su personaje y del color que habrá de caracterizarles durante el resto de la partida.

Cuatro colores cambian cíclicamente en la pantalla mientras el ordenador espera una señal de


perera.

disparo del joystick o una tecla; el más rápido, elegirá antes su personaje. Para elegir un personaje mediante el joystick, cualquier dirección de éste mostrará una especie en el centro de la pantalla; al oprimir disparo la elección será efectuada. En el teclado, la simulación del botón de disparo se verifica al pulsar dos teclas a un tiempo; para escoger, nos detendremos sobre el personaje situado a la izquierda del deseado, presionando después la tecla propia de subir ('↑' o 'Q') y a continuación la tecla de bajar ('=' o Commodore).



ficha técnica

NOMBRE:	M.U.L.E.
DISTRIBUIDOR:	DRO SOFT
PRECIO:	2.500 PTAS.
SOPORTE FISICO:	CINTA
INSTRUCCIONES:	COMPLETAS, TRADUCIDAS Y CON ALGUNAS ERRATAS
TIPO:	JUEGO DE TABLERO
OBSERVACIONES:	DE 0 A 4 JUGADORES, PRECISA AL MENOS UN JOYSTICK

Cada una de las ocho razas espaciales seleccionables posee su propia identidad y características. De acuerdo con los gustos del jugador, éste optará por el MECATRON, COGOTITA, DROCUMAN, GORIBOT, BOLOIDE, PLUMMOX, PATORIYO O HUMANO, teniendo siempre presentes las cualidades especiales de algunos de ellos:

El MECATRON es la especie seleccionada por el ordenador para suplir la carencia de cuatro jugadores humanos.

El record de falta de comedimiento para con la comida es ostentado por el BOLOIDE, cuyo nombre debiera haber sido el de «tragacosechas», en honor a la cantidad de alimentos que precisa.

Un PLUMMOX, especie novata, recibe al empezar el juego 600 $ extra y dispone de mayor tiempo en cada turno para poner a punto sus medios de producción.

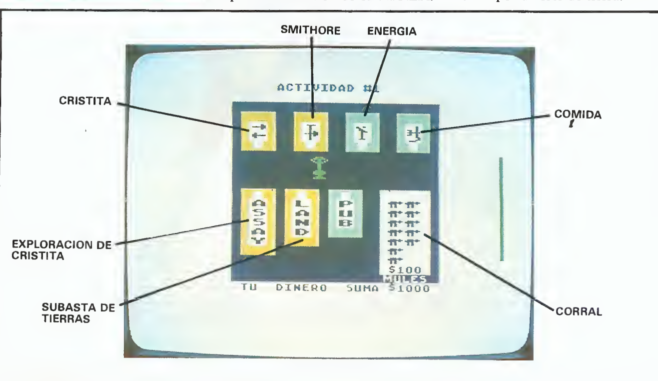
La inteligencia inherente a un HUMANO condiciona una reducción en su capital inicial de 300 $ y en el tiempo que se le concede para la instalación de su M.U.L.A.

LOS TURNOS DE JUEGO DE UNA PARTIDA: TODOS LOS PASOS QUE UN COLONO DEBE REALIZAR

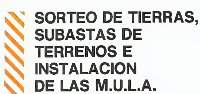
Posteriormente a la selección, el ordenador cargará la segunda parte del programa (versión en cinta) y, al finalizar, el proceso de colonización comenzará con la aparición en la pantalla del sumario, donde están representados, en orden de cuantía de su capital, los cuatro colonos y el dinero obtenido por cada uno hasta este momento. A partir de aquí, se sucederán los turnos de juego (6 ó 12) y sus respectivas fases.

En el nivel de NOVATO, las fases a cumplimentar son: sorteo de tierras, instalación de las M.U.L.A. y compra-venta de Smithore, Comida y Energía.

El nivel NORMAL, permite una nueva fase posterior a la del sorteo de tierras y anterior a la instalación de las M.U.L.A.: la subasta de terrenos, en la cual el ordenador o un jugador abre una subasta por un lote de tierra.



El último nivel, TORNEO, incluye todas las fases anteriores y añade un nuevo mercado de compraventa, el de Crystita. Además de ésto, ofrece la posibilidad de realizar CHANCHULLOS o actividades económicas efectuadas entre dos comerciantes, excluyendo momentáneamente al resto de los participantes del juego.



SORTEO DE TIERRAS, SUBASTAS DE TERRENOS E INSTALACION DE LAS M.U.L.A.

En el M.U.L.E., al igual que en el lejano Oeste, los conquistadores más rápidos obtienen primero su terreno, pues el ordenador va ofreciendo a los jugadores todos los lotes de tierra de la colonia sucesivamente, esperando una pulsación del botón de disparo o de una tecla. Todo comerciante tiene derecho a obtener un terreno por turno para desarrollar sus labores de producción, el cual será valorado en el sumario en 500 $, independientemente de su procedencia.

Las subastas de tierras, impracticables en el nivel NOVATO y efectuadas después del sorteo de terrenos, tienen por objeto la venta de un lote de tierra al mejor postor. Por lo general, nuestro Commodore desempeña el papel de vendedor, a menos que un jugador haya previsto una subasta en el turno precedente, entrando en la zona del pueblo denominada LAND.

La tierra, por sí misma, no produce nada; sólo con la ayuda de las M.U.L.A. es posible organizar monopolios de explotación de Crystita (100 $), Smithore (75 $), Comida (50 $) y Energía (25 $), en nuestras propiedades. En la fase de instalación de las M.U.L.A. tenemos a nuestra disposición un límite de tiempo para sentar lo medios de producción, cazar al Wampus y efectuar sondeos de Crystita.

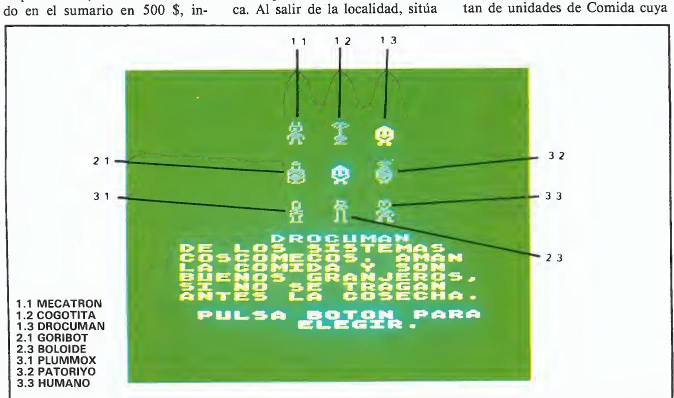
Nuestro representante en el Planeta Irata, manejado mediante el joystick, aparece en el centro del pueblo, donde adquiere una M.U.L.A. en el establo y la transforma para una actividad específica. Al salir de la localidad, sitúa su M.U.L.A. sobre la choza central de una de sus propiedades. Otras posibilidades son: cambiar la posición de una M.U.L.A. o reconvertirla para una nueva actividad en la población.

Si no deseas instalar otros artefactos, lo mejor es ir al «pub» y ganar dinero a cambio de tu tiempo restante. También resulta rentable la caza del Wampus, la cual será recompensada con 100 $; para ello, esperaremos a que abra su puerta en cualquiera de las montañas del tablero, nos situaremos encima y probaremos suerte con el botón de disparo. Una tercera posibilidad (aconsejable y muy lucrativa) la constituye el sondeo de Crystita verificable en el apartado de ASSAY, sito en el poblado.



LOS PRODUCTOS PROPIOS DE LA COLONIA Y SUS RESPECTIVOS MERCADOS DE COMPRA-VENTA

Los pioneros de Irata se alimentan de unidades de Comida cuya



1.1 MECATRON
1.2 COGOTITA
1.3 DROCUMAN
2.1 GORIBOT
2.3 BOLOIDE
3.1 PLUMMOX
3.2 PATORIYO
3.3 HUMANO



carencia produce la disminución en el tiempo de instalación de las M.U.L.A. de un modo drástico. El mejor terreno para producir este tipo de bienes es el río y su precio mínimo en los mercados de compra-venta es de 15 $. El 50 por ciento de la comida almacenada se pierde en cada turno.

Toda central de producción, debido a la actividad que desarrolla, consume energía suministrada por las factorías solares cuyo valor más bajo ronda los 10 $. El lugar más óptimo para el desarrollo de fuentes de energía es el llano, y su falta reduce el número de bienes producidos. La energía se deteriora en un 20 por ciento.

La minería también está presente en el tablero con la denominación de Smithore; no es imprescindible para la vida, pero supone una fuente de riqueza muy considerable. Su precio mínimo es de 14 $ y la zona idónea de fabricación son las montañas (cuantas más elevaciones haya en el terreno, más se producirá en él).

La Crystita, exclusiva del nivel de Torneo, se obtiene en aquellos terrenos que contengan vetas rentables, exceptuando el río, donde no puede instalarse ninguna M.U.L.A. de este mineral.

La cantidad máxima de este mineral almacenable es de 50 unidades, perdiéndose toda ganancia superior. Además, dos veces por partida, los piratas robarán el total de Smithore (NORMAL) o Crystita (TORNEO).

Todos estos productos obtenidos, entran en comercio durante las pantallas relativas a la compra-venta. Aquí, el ordenador informa acerca del exceso o defecto de cada jugador respecto al producto en cuestión y ofrece un margen de tiempo para que los jugadores puedan declararse como compradores o como vendedores en dicha actividad. El último paso consiste en manejar nuestro representante hacia arriba o hacia abajo para variar el precio de la transacción.

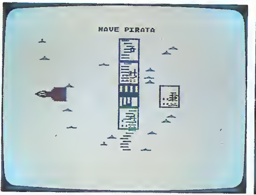
El nivel Torneo también admite aquí una peculiaridad denominada CHANCHULLO, merced a la cual dos jugadores que pulsen disparo a un tiempo (dos teclas en su defecto) dispondrán de un espacio temporal para comerciar entre ellos, sin intervención de los demás jugadores. Esta técnica se utiliza también cuando un colono decide vender su tierra a otro en la subasta de terrenos.

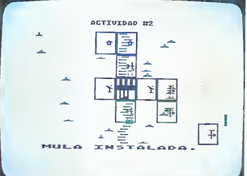


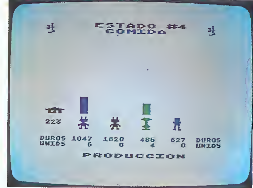
LAS BASES ECONOMICAS QUE RIGEN EL PRIMER ASENTAMIENTO COLONIAL DEL PLANETA IRATA

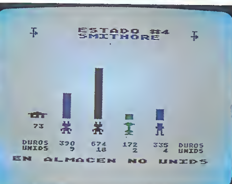
La economía del M.U.L.E. es tan espacial como especial; por ejemplo, tres terrenos que produzcan el mismo bien aumentan en una unidad su rendimiento, o dos

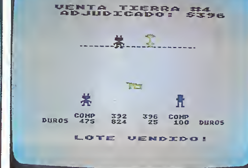












lotes de tierra colindantes aumentan sus resultados en una unidad.

De estos hechos, se deduce la conveniencia de obtener tierras de acuerdo con las necesidades propias, considerando las zonas más aptas para el desarrollo de las diversas industrias.

En la Tierra, el suelo que pisamos tiene un valor fundamental (al menos así lo creen las inmobiliarias), al giual que en el planeta Irata; cuantos más terrenos en funcionamiento existan, mucho mayor es la producción. Obviamente, debemos comportarnos como auténticos hombres de las finanzas en las subastas y no dejar NUNCA que un jugador obtenga un terreno por un precio inferior a los 500 $.

La cuestión del monopolio escogido depende directamente del grado de desarrollo de la partida. En los primeros momentos, urge conseguir comida y energía para sobrevivir, y en los últimos, el Smithore y la Crystita convertirán a cualquiera en un personaje rico.

La Comida debe venderse a precios muy altos en la compraventa y, caso de ser posible, evitar ayudar a los deficitarios humanos. El aspecto de Energía es esencial: si falta ésta no hay producción, luego evita vender cuando falte.

Smithore y Crystita son minerales cuyo comercio reporta grandes beneficios y será determinante en los últimos turnos de juego. Puede comprarse a bajo precio y vender a un precio superior, o si posees el total de Smithore y hay escasez de M.U.L.A., los precios se cuadruplicarán; así que suelta algunos aparatos para provocar su escasez.

Las asociaciones entre jugadores son tan temibles como la peste negra, puesto que pueden cerrar el comercio entre ellos y sacarte literalmente del juego. Sembrar cizaña entre asociados da unos resultados óptimos. A veces, unas palabras valen más que mil imágenes («te venderé lo que quieras», «confía en mí», «yo no se lo vendería», «cómpralo todo», etc.).

Por fin, para terminar, vale ya de perorata, recuerda que el vencedor será aquel cuyos nervios simulen el acero (no valen las aleaciones metálicas).

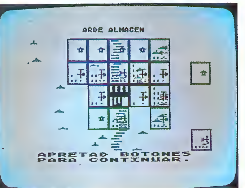
¡Esto es increíble! Uno de los escasos programas para ordenador verdaderamente originales, y distintos de todos los demás, asentado en una ciencia tan antigua como la economía.

Sin unos gráficos maravillosos, más bien mediocres, y un sonido hasta cierto punto aceptable, M.U.L.E. se constituye en una curiosidad del software para el Commodore 64.

Bajo una apariencia simple se oculta una programación complicada, de 3.500 horas según los autores, donde quedan reflejados multitud de conceptos propios de las industrias de producción.

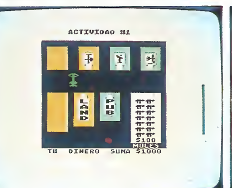
Aparte de algunos fallos, como la dificultad para detener el curso de juego con la barra espaciadora, el programa se presenta muy entretenido aunque transcurre con lentitud, subsanable mediante la pulsación del botón de disparo de joystick. Además, de principio a fin, contiene notas humorísticas que contribuyen definitivamente en la valoración positiva del programa.

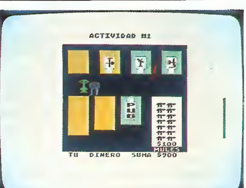












PROGRAMANDO AL MILISEGUNDO

No es necesario decir todo lo lenta que resulta la ejecución de un programa escrito en BASIC. Si lo que verdaderamente nos interesa es la rapidez, entonces nos hemos equivocado de lenguaje.

No obstante, hay ocasiones en las cuales la velocidad no juega un papel decisivo, y además, la comodidad de uso de un lenguaje de alto nivel (como lo es el BASIC), o la imposibilidad de programar en otro lenguaje, ya sea por la dificultad de aprendizaje o porque no lo tenemos disponible; hacen que, a pesar de todo, sigamos utilizando el archiconocido lenguaje.

El Basic de Commodore no se cuenta entre los más rápidos, si bien tampoco es el más lento. Pues bien, ya que a nadie le amarga un dulce, vamos a ver si conseguimos arañar esos milisegundos que siempre son vitales programando en este lenguaje.

A continuación daremos una serie de consejos para evitar ciertas costumbres, que sólo llevan a enlentecer, lo ya de por sí lento. El programa del listado, sirve para cualquier ordenador Commodore, consta de una serie de 9 ejemplos que nos aclararán lo expuesto a continuación. Los ejemplos se ejecutarán con **RUN 100, RUN 200,** etc... y nos darán un número que indica el tiempo de ejecución en sesentavos de segundo. En cada caso se dan las modificaciones necesarias para ver la diferencia.

EMPLEAR VARIABLES EN LUGAR DE CONSTANTES AHORRARA UN TIEMPO PRECIOSO EN LA DECODIFICACION

El primer consejo, es que usemos variables en vez de constantes. Si el intérprete Basic se encuentra con una constante, lo primero que debe hacer es decodificarla, lo cual implica el uso de una rutina de conversión que no es muy rápida; en cambio, si se encuentra con una variable, la decodificación ya ha tenido lugar en el momento de la asignación, y sólo tiene que buscar su valor en la memoria.

Para probar esto ejecutaremos **RUN 100** y anotaremos el resultado que aparece en la pantalla, a continuación modificaremos la línea 130 como sigue: **130 POKE53280,I.** Volvamos a ejecutar **RUN 100** y obtendremos la diferencia de tiempo. En el primer caso hemos empleado una variable, mientras que en el segundo utilizamos una constante.

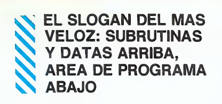
En el segundo caso, vemos simplemente la diferencia que hay al usar una línea multisentencia; es decir, introducir en una línea tantas instrucciones como sea posible. Si bien esto no es lo más claro y estético, en nuestro caso perseguimos la velocidad, así que...; con **RUN 200** tendremos el nuevo valor comparado con el del ejemplo anterior.

Otro factor que influye notablemente en la velocidad de ejecución, son las líneas REM; están muy bien para la fase de preparación, pero después ¡¡fuera con ellas!!. Por tanto, otra costumbre a eliminar es usarlas como líneas «blanco» o de destino en GOTO, GOSUB, etc... Para probar esto ejecuta **RUN 300**, apunta el valor y borra la línea 320; vuelve a ejecutarlo y ahí tienes la diferencia.



EL PUNTO DECIMAL: UN EXCELENTE SUSTITUTIVO PARA LA CONSTANTE CERO

La prueba que viene ahora, es un tanto curiosa: si tienes que efectuar una puesta a cero de una variable (en nuestro caso una matriz dimensionada a 999), no uses el valor 0, usa en cambio el punto decimal «.». Esto es algo parecido a lo que decíamos en nuestro primer ejemplo. **RUN 400** nos dará el resultado «corto», cambia la línea 420 a: **420 X(I)=0** y a continuación vuelve a ejecutar **RUN 400.** ¿Convencido?



EL SLOGAN DEL MAS VELOZ: SUBRUTINAS Y DATAS ARRIBA, AREA DE PROGRAMA ABAJO

En el ejemplo número 5, vamos a ver como el intérprete BASIC busca las direcciones de salto (**GOTO, GOSUB**), empezando por la primera línea de BASIC, es decir, recorre todo el programa hasta que la encuentra. Con **RUN 500** veremos la opción más lenta, para «apretar el acelerador», cambia la línea 520: **520 GOSUB 1.**

Con las líneas **DATA,** sucede exactamente lo mismo; aquí la mayoría estamos mal acostumbrados, pues a nadie se le ocurre poner las «DATAS» al principio de un programa; pero para ver la diferencia, tras apuntar el resultado obtenido con **RUN 600,** copia la línea 1001 en la línea 2 (**2 DATA 4**) y borra la 1001. Ejecuta nuevamente **RUN 600** y... Conclusión: Las subrutinas y las «DATAS» ¡al principio! No conviene olvidar que todo esto es para ganar velocidad, si nos da lo mismo un minuto que dos, lo pondremos como más nos guste.

En la línea 720, vemos una construcción un tanto extraña de IF...THEN; en principio es equivalente a: **720 IF I = 999 AND A = 0 THEN PRINT "FIN",** sin embargo, prueba esta última modificación después de ver el resultado de la original (**RUN 700**). La explicación es bien simple, **AND** es difícil de evaluar, y en el segundo caso se deben comprobar las dos condiciones (I=999 y A=0) para cumplir; mientras que en el primer caso vemos si I=999 y si no lo es... ¡fuera!.

LOS NOMBRES LARGOS DE VARIABLES SON TAN INUTILES COMO LENTOS

Con RUN 800 veremos el tiempo que perdemos al usar nombres de variables demasiado largos. Lo mejor es utilizar una o dos letras, dado que el ordenador sólo reconoce los dos primeros caracteres de las variables. Cambia la línea 820 así: **820 A=1111** y verás los resultados.

Y POR ULTIMO, SAQUEMOSLE TODO EL PARTIDO POSIBLE A LA ALTA VELOCIDAD DE PRINT

Al dar RUN 900, vemos la rapidez de **PRINT**; aprovechémosla siempre que podamos en construcciones como las de la línea 920. Para comprobar esto ejecutemos de nuevo **RUN 900** con las siguientes modificaciones:

910 FORI=11TO333 y
920 PRINT"AAA";
910 FORI=1TO999 y
920 PRINT"A";

Aplicando todo lo anterior, podemos descubrir como nuestros programas aumentan perceptiblemente de velocidad, aunque no de forma espectacular; pero ya se sabe: tacita a tacita...

```
1 RETURN                                          -049-
100 TI$="[6 0]"                                   -019-
110 X=53280                                       -073-
120 FORI=0TO255                                   -143-
130 POKEX,I                                       -176-
140 NEXT                                          -244-
150 PRINTTI:END                                   -241-
200 TI$="[6 0]"                                   -020-
210 X=53280:FORI=0TO255:POKEX,I:NEXT:PRINTTI:END
                                                  -132-
300 TI$="[6 0]"                                   -021-
310 FORI=0TO999                                   -159-
320 REM                                           -153-
330 NEXT                                          -245-
340 PRINTTI:END                                   -242-
400 TI$="[6 0]"                                   -022-
410 DIMX(999):FORI=0TO999                         -008-
420 X(I)=.                                        -019-
430 NEXT                                          -246-
440 PRINTTI:END                                   -243-
500 TI$="[6 0]"                                   -023-
510 FORI=0TO999                                   -161-
520 GOSUB1000                                     -248-
530 NEXT                                          -247-
540 PRINTTI:END                                   -244-
600 TI$="[6 0]"                                   -024-
610 FORI=0TO999                                   -162-
620 READA:RESTORE                                 -115-
630 NEXT                                          -248-
640 PRINTTI:END                                   -245-
700 TI$="[6 0]"                                   -025-
710 FORI=0TO999                                   -163-
720 IFI=999THENIFA=0THENPRINT"FIN"                -194-
730 NEXT                                          -249-
740 PRINTTI:END                                   -246-
800 TI$="[6 0]"                                   -026-
810 FORI=0TO999                                   -164-
820 AAAAAAA=1111                                  -130-
830 NEXT                                          -250-
840 PRINTTI:END                                   -247-
900 TI$="[6 0]":PRINT"[CLR]";                     -244-
910 FORI=1TO111                                   -142-
920 PRINT"[9 A]";                                 -016-
930 NEXT                                          -251-
940 PRINTTI:END                                   -248-
1000 RETURN                                       -193-
1001 DATA 4                                       -080-
```

CONCURSO

El travieso C-Byte tiene el honor de invitaros a la participación en nuestro primer concurso de programación. Los requisitos necesarios son bien pocos:

- Saber programar un ordenador COMMODORE.
- Ser español o extranjero y
- Tener una edad comprendida entre 5 y 105 años.

Fácil, ¿verdad?
En cuanto a los premios, la mar de atractivos:

- **1.er premio.–60.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **2.° premio.–30.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **3.er premio.–15.000 pesetas en material informático a escoger por el premiado, más un poster de C-Byte.**

Y en fin, si alguno de los programas destaca por su originalidad, estética o comicidad, no sería de extrañar que le cayera alguna cosilla más...

Las bases del concurso son bien sencillas:

1. Los programas remitidos al concurso deberán ser creación original del autor o autores, y completamente inéditos, pudiendo remitir tantos programas como se desee.

2. Los programas deberán ser enviados en casette o disco flexible a TU MICRO COMMODORE (Concurso de programación). Apartado de correos 61.294. 28080 MADRID.

3. Los programas podrán ser de cualquier tipo (juegos, utilidades, gestión, educativos) y habrán de estar escritos en lenguaje BASIC o código máquina.

4. Los programas deberán ser remitidos desprovistos de cualquier tipo de protección, que impida o dificulte el análisis del mismo, así como reproducción en estas páginas y su introducción como listado siguiendo el sistema FUERA ERRORES.

5. Cuando la ejecución del programa precise de la concurrencia de determinado periférico o aditamento (joysticks, tabletas gráficas, programas comerciales de ayuda), se valorará la indicación de las modificaciones pertinentes, para que el programa pueda ser disfrutado por cualquier usuario en la configuración básica.

6. Todo programa presentado al concurso deberá acompañarse de los siguientes datos:
   - Datos personales del concursante.
   - Nombre del programa.
   - Modelo para el que está destinado.
   - Breve descripción del programa detallando las indicaciones necesarias para su ejecución.

7. Los programas premiados pasarán a ser propiedad de la revista TU MICRO COMMODORE, pudiendo hacer ésta libre uso de ellos, y renunciando sus autores a cualquier otra compensación distinta al premio.

8. Los programas no premiados que por su calidad se hagan merecedores de su publicación, serán adquiridos por la editorial, aplicando la tarifa vigente.

9. Los programas recibidos con posterioridad a la fecha tope de la presente edición, serán automáticamente incluidos en los destinados a la siguiente.

10. El jurado decidirá sobre todos los aspectos no contemplados en estas bases y su decisión será inapelable.

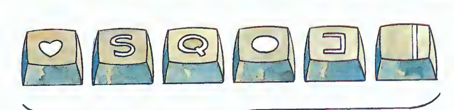
Y ahora a darse mucha prisa, el plazo para la recepción de programas termina el próximo día 15 de marzo.

¡¡SUERTE!!



CURSO DE BASIC

Ya hemos adquirido una pequeña experiencia en el desplazamiento del cursor, aplicada a la modificación de líneas de programa. Sabemos también acerca de las funciones de las teclas **HOME** y **CLR HOME,** destinadas a la vuelta al origen del cursor con o sin borrado de la pantalla. Sin embargo, la utilización de estas funciones de posicionamiento libre en un punto cualquiera de la pantalla, nos han estado reservadas a



TECLAS DE CONTROL DE PANTALLA

El comportamiento de las teclas de desplazamiento del cursor y control de la pantalla al codificar líneas de instrucción, es el habitual de posicionamiento



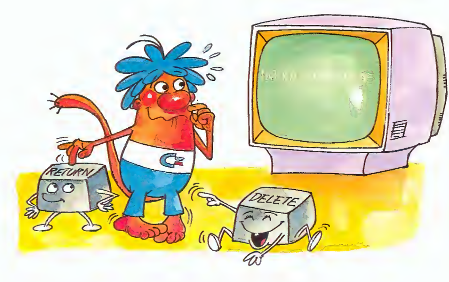
Cuando queramos salir del efecto de almacenamiento como caracteres de control de las teclas de desplazamiento, basta con pulsar las dobles comillas.

la acción en comando directo, es decir, siempre fuera de programa.

Pues bien, nada más fácil. El COMMODORE 64 permite incluir en forma de **string,** como literal o variable alfanumérica, caracteres de control relativos a estos desplazamientos.

Cabría preguntarnos, sin embargo, como es capaz el ordenador de distinguir cuando nos encontramos introduciendo un programa, entre los desplazamientos del cursor que producimos para acceder a determinadas posiciones de una línea, y los caracteres que deseamos sean incluidos para su posterior ejecución. Para ello, este se basa en la pulsación de la tecla de las dobles comillas (").

Cuando nos encontramos codificando líneas de instrucciones, el comportamiento de las teclas de desplazamiento del cursor y control de la pantalla es el habitual, es decir, cada pulsación produce el efecto inmediato de posiciona-



Para abandonar la situación producida al borrar con la tecla **DELETE** *una de las dobles comillas, podemos utilizar* **RETURN.**

REPRESENTACION DE LOS CARACTERES DE CONTROL		
NOMBRE DE LA FUNCION	SHIFT	REP
CURSOR ORIGEN	SIN	" S "
CURSOR ORIGEN (BORRANDO)	CON	" ♥ "
CURSOR ABAJO	SIN	" Q "
CURSOR ARRIBA	CON	" ● "
CURSOR DERECHA	SIN	"] "
CURSOR IZQUIERDA	CON	" | "

miento. Sin embargo, a partir de la pulsación de la tecla de las comillas dobles ("), el ordenador representa cada pulsación de alguna tecla de posicionamiento del cursor en forma de carácter de control en video invertido, de modo que queda almacenado en forma de cadena de caracteres, no siendo ejecutado el posicionamiento de forma inmediata.

Como medio para salir de este efecto de «almacenamiento» como caracteres de control de las teclas de desplazamiento, basta con pulsar nuevamente las comillas dobles ("), indicativas del final de la definición del literal, o la pulsación de la tecla **RETURN,** indicativa del final de la instrucción.

Este sistema de conexión/desconexión por pulsación de las comillas dobles (") es muy efectivo y simple. Sin embargo, debemos tener en cuenta que el ordenador lleva un control, dentro de cada línea de instrucción, del número de comillas pulsado para saber si resulta par o impar, procediendo en consecuencia a la ejecución inmediata del desplazamiento o al almacenamiento del mismo, respectivamente.

En este estado de cosas, es evidente que «confundimos» al ordenador cuando borramos una comilla con la tecla **DELETE,** y por ello nos podemos ver envueltos en una situación embarazosa de la cual, no obstante, podemos salir airosos con **RETURN** y volviendo a entrar en la línea o pulsando nuevamente las comillas y borrándolas a continuación.

Debemos saber además que, cuando nos «abrimos espacio» dentro de una línea de instrucción mediante la pulsación de la tecla **INSERT** (**SHIFT** + **INST/DEL**), la serie de espacios en blanco generados son considerados por el ordenador como si de «zona de comillas» se tratara. Por ello, debemos tener en cuenta que el hecho de avanzar con cursor derecha por este «hueco» producirá la impresión en pantalla del correspondiente símbolo del carácter de control además de ejecutar la acción de avanzar el cursor una posición.

Superar este pequeño inconveneinte no es difícil. Podemos optar por avanzar con la barra espa-

En el programa de desplazamiento del móvil, el espacio en blanco que precede a la impresión del carácter, está destinado a borrar la anterior posición que ocupaba.



ciadora, escribir directamente los caracteres que deseamos para modificar el contenido de la instrucción, e incluso, utilizar en último término la tecla **DELETE** (**INST/DEL**) para eliminar cualquier carácter no deseado, antes de proceder a la validación definitiva de la instrucción con **RETURN.**

LOS CARACTERES DE CONTROL

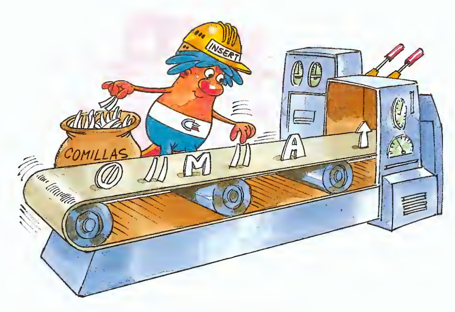
Existen seis caracteres de control posibles que se corresponden con la pulsación, en el modo comillas, de las dos teclas de desplazamiento del cursor y la de control de la pantalla, con y sin **SHIFT.** Su efecto y representación son los representados en la tabla adjunta.

La combinación que hagamos de estos caracteres de control es completamente libre, dependiendo del posicionamiento deseado dentro del programa. A modo de ejemplo, veamos la forma de escribir un mensaje en el centro de la pantalla:

```
10 PRINT " QQQQQQQQQQQQ]]]
]]]]]]]]]]]COMMODORE 64"
```

Antes que nada, borramos la pantalla y nos posicionamos en su origen (ángulo superior izquierdo) por medio de **SHIFT** y **CLR HOME.** A continuación, colocamos 12 cursores abajo para posicionarnos en la línea central de la pantalla (la 12 contando de 0 24). Por último, incluimos 14 cursores derecha para centrar el mensaje dentro de la línea, ya que este ocupa 12 posiciones, y escribimos el texto en cuestión.

Realmente, nos es absolutamente igual efectuar el desplazamiento horizontal por medio de cursores a la derecha o de la barra espaciadora, puesto que no nos encontramos imprimiendo sobre nada ya existente en la pantalla. Sin embargo, en ocasiones, resulta interesante tener la posibilidad de «sobreimprimir». En este programa tenemos un ejemplo claro:



Al pulsar la tecla ***INSERT****, la serie de espacios generados son considerados como zona de comillas.*

```
10 PRINT ""
20 FOR I=1 TO 920:PRINT"*";:
NEXT
30 PRINT "SQQQQQQQQQQQ
Q]]]]]]]]]]]]]]COMMODORE 64"
```

En este caso, borramos previamente la pantalla en la instrucción 10 para posteriormente escribir 23 líneas de asteriscos (23 × 40 = 290) por medio de un bucle **FOR-NEXT**. Por último, escribimos el mensaje anteriormente propuesto con la particularidad de que esta vez el carácter de comienzo de la cadena de impresión es el de posicionamiento en el origen sin borrado.

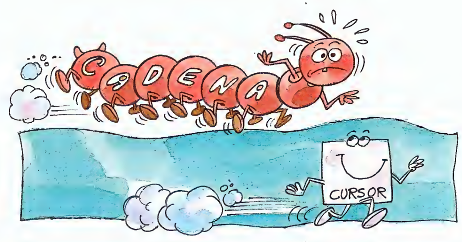
SIMULACION DE MOVIMIENTO

Otra de las características del control del cursor dentro de programa es el hecho de permitir el desplazamiento de móviles. El siguiente programa produce el desplazamiento de una pelota que rebota una y otra vez en cada una de las paredes de la pantalla a gran velocidad:

```
10 PRINT "QQQQQQQQQQQQ";
20 FOR 1 = 0 TO 38
30 PRINT "*";
```

Si queremos que el programa de desplazamiento del móvil marche a toda velocidad, basta con suprimir las instrucciones 40 y 80.





Podemos incluir en nuestro C-64, en forma de cadena, los caracteres de control para desplazamientos del cursor en la pantalla

```
 40 FOR J = 0 TO 19:NEXT
 50 NEXT
 60 FOR I = 0 TO 38
 70 PRINT "*";
 80 FOR J = 0 TO 19:NEXT
 90 NEXT
100 GOTO 20
```

En la línea 10 efectuamos el borrado de la pantalla y posicionamiento inicial al comienzo de la línea central de la misma.

Las líneas 20 a 50 tienen por objeto hacer avanzar el móvil de izquierda a derecha. Debemos tener en cuenta que el espacio en blanco que precede a la impresión del carácter está destinado a borrar la anterior posición que ocupaba éste. Por ello, no podemos sustituirlo por un carácter de cursor a la derecha. De hacerlo, la pelota se desplazaría dejando una «estela» tras de sí, del mismo modo que si únicamente incluyeramos el carácter de la pelota entre las comillas. Además, pensemos en que después de producirse cada una de las impresiones de la pelota, el cursor se posicionará de forma automática a la derecha del carácter escrito, como es habitual. Para controlar esta situación, debemos incluir un carácter de cursor izquierda después de la impresión.

Las líneas de 60 a 90 producen el efecto de rebote, para hacerlo empleamos los caracteres de control del cursor atrás doblemente, con lo que cada móvil impreso lo es una posición más a la izquierda que el anterior, sin olvidarnos de comenzar la cadena de impresión con un espacio en blanco de borrado.

Por último, en la línea 100 incluimos un **GOTO** imperativo a la línea 20, para conseguir que el efecto se repita indefinidamente.

En este programa podemos ver también una curiosa aplicación de bucle **FOR-NEXT** destinado a retardar únicamente la ejecución del programa. Es el caso de las líneas 40 y 80, donde incrementamos sin ningún objetivo el valor de la variable J desde 0 a 19.

Si deseamos que el programa marche más lentamente para apreciar mejor el movimiento, basta con modificar el valor 19 a alguno superior. Si por el contrario deseamos una ejecución a toda velocidad, podemos optar por suprimir las instrucciones 40 y 80, escribiendo simplemente su número y tecleando **RETURN**. Puede ser, además, una oportunidad de ver lo rápidamente que trabaja el BASIC en lo que se refiere al tratamiento de la pantalla.

En general, podemos decir que hemos construido un programa simple que ejecuta una acción determinada haciendo uso de dos bucles anidados **FOR-NEXT** y un tercer bucle indefinido marcado por el **GOTO** incondicional de la línea 100. El programa, por lo tanto, no tendrá fin; siendo necesario pulsar la tecla **RUN/STOP** para interrumpir la ejecución del mismo.

Estamos ahora en condiciones de proponer un ejercicio práctico que puede consistir, apoyándonos en los desplazamientos del cursor, en codificar un programa que imprima caracteres gigantes en la pantalla, letra a letra.

Para hacerlo, podemos diseñar primero cada uno de los caracteres diferentes a emplear sobre una hoja de papel cuadriculado, empleando por ejemplo, una matriz de 7 filas por cinco columnas, lo cual nos permitiría escribir mensajes de hasta 6 caracteres de ancho escritos en tres líneas.

Hecho ésto, basta ya con tener la paciencia necesaria para reproducir exactamente los movimientos que nosotros haríamos para escribir estos caracteres manualmente sobre una hoja de papel, incorporando el correspondiente retardo entre la impresión de cada carácter.



Para codificar un programa que utilice caracteres gigantes, el primer paso es el diseño sobre el papel cuadriculado de los mismos.

BOULDER DASH

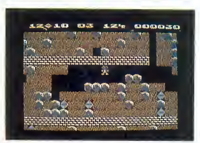
Rockford, hábil personaje, se halla en un laberinto de estancias y cuevas, para salir de las cuales deberá recoger todas las joyas posibles, escondidas en los recovecos de las habitaciones y defendidas por numerosos peligros que iremos descubriendo.

En total existen 16 cuevas, cada una de ellas comprendida en varias pantallas, ordenadas de A a P. Así, si logramos acabar por completo Boulder Dash, habremos pasado previamente por 256 habitaciones (¡fiuuu!).

Cada pantalla de los niveles es diferente, y en cada una de ellas habremos de ayudar a Rockford a realizar una distinta misión. En la A, deberemos coger todas las joyas que podamos antes de que el tiempo se agote y encontrar la puerta de salida. La B es parecida, sólo que para conseguir recoger las suficientes gemas habremos de mover pesadas rocas que obstruirán nuestro camino, pudiendo aplastarnos si no ponemos cuidado. En otras, como la C, tendremos un tiempo limitado para encontrarlas todas. Las pantallas van creciendo en dificultad y por ello, a medida que vayamos superándolas nos enfrentaremos a los insectos de fuego, mariposas o la creciente ameba.







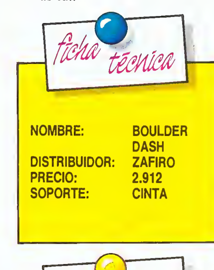


Este diabólico juego tiene sus propias reglas, así las piedras caerán según la ley de la gravedad... pero la de Boulder Dash; estas reglas sólo las podremos aprender con la práctica del juego. Los insectos de fuego son mortales y se deslizan por los túneles que Rockford va construyendo; la única defensa contra ellos es hacer que exploten, dejando caer piedras sobre ellos, aunque ésto no es nada fácil, como podréis comprobar.

Las mariposas de colores tienen un comportamiento parecido al de los insectos de fuego, pero por el contrario, éstas al explotar se convertirán en preciosas joyas que podremos acarrear. La creciente ameba es un pequeño virus que nace en una gota verde, pero que se va reproduciendo y creciendo en el aire y tierra. La única forma de detenerla es rodear todo su posible espacio de crecimiento por rocas, que lograrán sofocarla; cuando esto ocurra, todos los sectores ocupados por la ameba se convertirán en joyas.

Otro componente curioso de este juego es el muro encantado, a primera vista igual a cualquier otro que rodea a Rockford, pero con una diferencia: cuando es golpeado por una piedra comienza a vibrar, y a partir de este momento, cualquier roca que caiga sobre el se convertirá en joya.

Estas y otras muchas son las curiosidades que nos encontraremos en el extraño mundo de Boulder Dash, el resto ¡tendrás que descubrirlas tú!.





TREASURE ISLAND



NOMBRE:	TREASURE ISLAND
DISTRIBUIDOR:	POWER
PRECIO:	2.600 PTAS.
SOPORTE:	CINTA









Cuando Robert Louis Stevenson creó de su imaginación el fabuloso libro «La Isla del Tesoro», quizás nunca se figuró el posterior éxito que éste iba a conseguir; por ello, esta agradable historia ha llegado incluso a ser trasladado a nuestro C-64.

El juego en sí consta de varias bases, donde Jim Hawkins (el «prota» de la novela) deberá sortear todos los peligros y situaciones adversas que se le presenten, que serán numerosas, puesto que la codicia impulsa a muchos siniestros personajes a luchar por el tesoro escondido.

En primer lugar, Jimmy debe lograr introducirse en el barco «Hispaniola», pero no será fácil, dado que Paw «el ciego» se lo intentará impedir, haciendo rápidas rondas junto a la pasarela de embarque.

Una vez dentro, Jimmy ha escuchado por casualidad los planes de amotinamiento de la tripulación y deberá saltar de barril en barril para librarse de los fuertes brazos que intentan atraparle para acabar con él, puesto que si nuestro héroe consigue contarle al capitán lo que sabe, sus planes se irán al traste.

Si hemos logrado sortear positivamente todos estos impedimentos, empieza verdaderamente la gran aventura. Jimmy consigue por fin desembarcar en la isla, junto con el doctor, el capitán, y otros marineros que no se han unido al motín. Después de construir una empalizada en el centro de la isla, son atacados por los piratas, pero en mitad de la lucha, Jimmy consigue escapar y se dispone a buscar el tesoro.

Primero debe encontrar la cueva de Benn Gunn, extraño personaje abandonado a su suerte muchos años atrás, cuando el tesoro fue enterrado en la isla. Una vez realizado este propósito, Jim buscará el tesoro por la isla, con la ayuda de un plano que divide ésta en 64 secciones. Una vez descubierto el tesoro, y después de haber salvado ataques de los piratas y animales, Jim deberá llevarlo al barco, que está anclado en la ensenada del «espía de cristal». Esta es la última fase: si es llevada a cabo, habremos finalizado con total éxito la aventura, lo cual no es en absoluto fácil pues la isla está recorrida por sinuosos y traicioneros caminos, a la vuelta de cada cual podremos encontrarnos con un fatal desenlace.

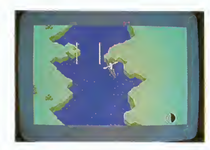
¡Animo!,... ¡y a la búsqueda del tesoro!

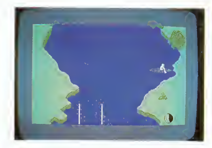


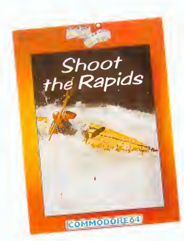
GRAFICOS:	***
SONIDO:	***
DIFICULTAD:	****
ORIGINALIDAD:	*****
INTERES:	****

SHOOT THE RAPIDS











NOMBRE:	SHOOT THE RAPIDS
DISTRIBUIDOR:	ZAFIRO
PRECIO:	2.300 PTAS.
SOPORTE:	CINTA

Las aguas rápidas nos arrastran hacia los peligrosos extremos del río, llenos de piedras que pueden hacernos zozobrar fácilmente, nuestros brazos se hallan entumecidos debido al elevado esfuerzo a que nos estamos sometiendo, y nuestro cerebro embotado por los golpes de agua fría, pero aun así sabemos que debemos seguir: la meta está cerca; perder la moral en este momento sería nuestro fin de competición.

Nuestro objetivo es, desde la salida, llegar en el menor tiempo posible a la meta, situada unos metros más abajo en el río, pero este trazado es muy peligroso dado que las aguas están muy embravecidas y nos impiden maniobrar con facilidad nuestra canoa a través del río. Como si la dificultad fuera pequeña, aún hay más, ya que simultáneamente tendremos que ir rebasando puertas colocadas en el recorrido; si no lo hacemos, seremos penalizados.

El tiempo aparece impreso en la derecha de la pantalla, en forma de reloj que va agotando sus minutos. Si no logramos acabar antes de la finalización del tiempo, quedaremos irremisiblemente descalificados, debiendo empezar de nuevo si queremos competir otra vez.

Shoot the Rapids está realizado con total realismo, por lo cual,



GRAFICOS:	***
SONIDO:	**
INTERES:	***
ORIGINALIDAD:	****
DIFICULTAD:	*****

cuando movamos la paleta derecha de nuestro hombre con el joystick, éste se desviará hacia la izquierda, y viceversa. Por ello, si no queremos desviarnos excesivamente de nuestro curso, deberemos llevar un ritmo continuado en ambos remos, lo cual no será fácil desde un principio. Intentar correr más no será bueno, puesto que las paletadas no entrarán suficientemente en el agua y no nos impulsarán todo lo que sería de desear; así, busquemos un ritmo adecuado a nuestras pretensiones y conforme vayamos dominando más el recorrido podremos ir subiendo éste paulatinamente.

Sobre todo, habrá que poner especial atención en los cambios de velocidad del río, puesto que las aguas en determinadas partes serán más rápidas que en otras, y si nos impulsan contra un lado del río y golpeamos lateralmente nuestra canoa, el hundimiento será seguro. Para maniobrar mejor tenemos la facultad de remar hacia atrás, pulsando el botón de disparo.

Pon tus músculos a tono, y prepárate porque empieza ya... ¡Shoot the Rapids!



2. WIZARDRY



3. KARATEKA





Colaboremos todos en la confección de estos 7 MAGNIFICOS de TU MICRO COMMODORE. Envíanos el nombre de tu programa favorito, dejando muy claro tu nombre y dirección. Todos los meses, sortearemos una suscripción por un año a nuestra revista entre las respuestas recibidas. Anímate y escríbenos a: TU MICRO COMMODORE. (7 MAGNIFICOS). APARTADO DE CORREOS 61.294. 28080 MADRID.



ESTA SEMANA

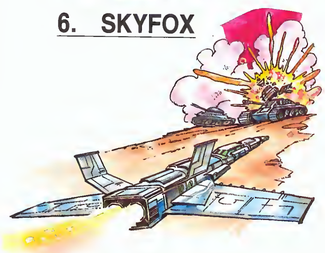
1. IMPOSSIBLE MISSION

Para producir una explosión nuclear, el malvado Dr. Elvin ha conseguido las claves de seguridad de las terminales de ordenadores de las grandes potencias del mundo, disponiendo así de toda la fuerza atómica existente. Gracias a nuestros conocimientos informáticos, nos hemos introducido en los procesadores de datos de Elvin y averiguado la hora del lanzamiento de los misiles atómicos. Nosotros: el agente 4125 (hace tiempo que 007 y muchos otros pasaron a mejor vida), haciendo frente a los pérfidos robots del loco doctor, debemos evitar la catástrofe.

4. RAMBO



6. SKYFOX



5. WINTER GAMES



7. ARCHON



SU MAJESTAD EL M

El microprocesador es sin duda el elemento más importante del ordenador, sobre todo a la hora de programar. Véamoslo en detalle.

Antes de comenzar, debemos hacer una advertencia. Los microordenadores Commodore emplean distintos microprocesadores de la familia 6500. Las diferencias entre ellos son mínimas, de modo que la práctica totalidad de lo estudiado en esta sección será válido para todos ellos. De cualquier forma, mencionaremos las diferencias entre los distintos tipos.

Una vez hecha esta aclaración, pasemos directamente al tema de este capítulo. El microprocesador más conocido de la serie 6500 es sin duda el 6502. Este equipaba a los viejos PET, Apple II, BBC y VIC, por lo que ha conquistado un puesto privilegiado en la historia de la microinformática.

LA VALVULA FUE EL PRINCIPIO DE TODO, AUNQUE GRANDES DESCUBRIMIENTOS LA SUCEDIERON

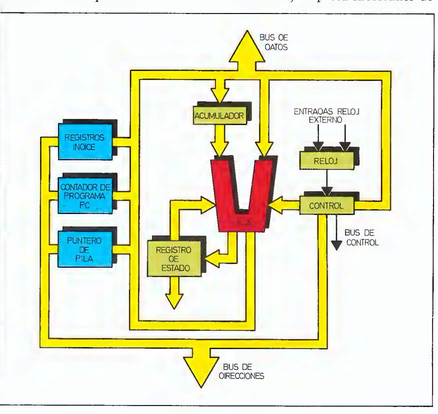
Los primeros ordenadores eran desmesuradamente grandes, de un tamaño equivalente al de una planta entera de un edificio. El consumo de energía era proporcional a su tamaño: se cuenta que, cuando uno de aquellos prototipos funcionaba, las luces del oeste de la ciudad sufrían una disminución considerable...

El cambio de la válvula de vacío por el transistor fue el primer caso hacia lo que hoy conocemos como ordenador. Se redujo drásticamente el tamaño, consumo e infiabilidad del sistema, y con ello el coste. Sin embargo, éste era aún excesivo para muchas aplicaciones, y era evidente que se hacía necesario aún un gran avance.

A finales de los años 50, los investigadores descubrieron el proceso para la producción de circuitos integrados, es decir, circuitos completos construidos sobre una pastilla de silicio semiconductor. Posteriores avances permitieron incluir cada vez más componentes en menos espacio, hasta llegar a miles de transistores en 2 mm. cuadrados de superficie.

La compañía norteamericana INTEL (curiosamente, a raíz de un encargo de una empresa japonesa) desarrollo en 1969 el 4004: el primer microprocesador de 4 bits; seguido poco después por el 8008, de 8 bits. El boom de estos circuitos es algo de lo que todos hemos oído hablar.

La lucha entre las distintas compañías fue épica; nuevos modelos sucedían a los anteriores, con mayor calidad y menor coste. La empresa MOTOROLA desarrolló el 6800, primer micro de una familia de gran potencia. Un grupo de ingenieros escindidos de esa compañía, desarrolló un chip basado en el anterior, pero simplificado y dotado de algunas cualidades superiores. De la unión de este equipo con la MOS TECHNOLOGY, empresa fabricante de



ICROPROCESADOR

circuitos para calculadoras, nació la familia 6500, de gran éxito en el mercado.



LOS GRANDES SECRETOS ENCERRADOS EN UNA PEQUEÑA CAJITA DE PLASTICO NEGRO

Un esquema simplificado de un microprocesador lo podemos ver en la figura 1: Los conceptos de bus ya nos son familiares, por lo que describimos someramente los restantes elementos.

El registro de control es, digamos, el director de la orquesta de la CPU. Se encarga de controlar el funcionamiento del sistema, generando las señales de sincronismo para todas las operaciones y para el flujo de datos hacia y desde el exterior.

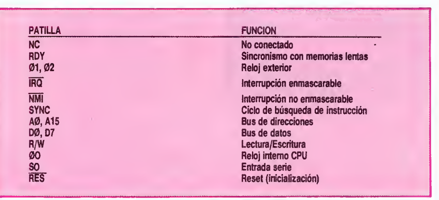
La unidad de control está dirigida por los impulsos de un reloj interno, que a su vez está controlado por un reloj exterior. El término de reloj se refiere a un dispositivo que genera impulsos de tensión a intervalos regulares. Cada vez que la unidad de control detecta uno de estos impulsos, envía una de sus señales de operación.

La ULA (Unidad Lógico-Aritmética) es la calculadora del microprocesador. Ella realiza todas las operaciones aritméticas (suma y resta) y lógicas (AND, OR, etc.). Sus salidas se dirigen al bus de datos, y como veremos, a registros auxiliares que indican otros datos sobre la operación: si se ha producido acarreo, si el resultado es cero o negativo, etc. Este registro de estado es de gran utilidad, como tendremos oportunidad de comprobar más adelante.

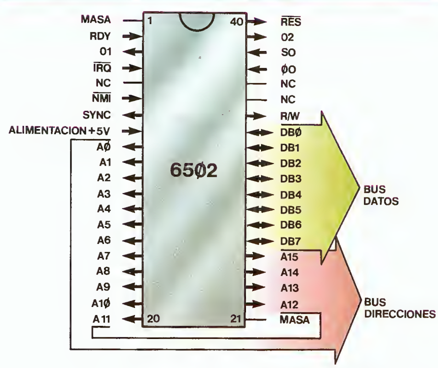
Los buses, como ya sabemos, son las líneas encargadas de llevar la información entre los distintos elementos del microprocesador. El bus de datos consta de 8 bits, y el de direcciones de 16, lo cual nos da acceso a 64 K de memoria. El 6510 tiene además unas líneas adicionales que le permiten acceder a 84 K, aunque no simultáneamente.

Es decir, podemos elegir entre ciertas direcciones, un bloque de memoria u otro; por ejemplo, escoger la ROM o la RAM de la zona de Basic, etc. Esto no es más que un truco para poder acceder a más de 64 K con un bus de 16 bits.

Los registros del micro, la parte que más interesa al programador, serán analizados en profundidad en el próximo capítulo de esta serie.



PATILLA	FUNCION
NC	No conectado
RDY	Sincronismo con memorias lentas
Ø1, Ø2	Reloj exterior
$\overline{IRQ}$	Interrupción enmascarable
$\overline{NMI}$	Interrupción no enmascarable
SYNC	Ciclo de búsqueda de instrucción
AØ, A15	Bus de direcciones
DØ, D7	Bus de datos
R/W	Lectura/Escritura
ØO	Reloj interno CPU
$\overline{SO}$	Entrada serie
$\overline{RES}$	Reset (inicialización)



ARCO

Como auténticos Guillermo Tell de salón, nos veremos practicando un deporte tan bello como antiguo en la historia del hombre: el tiro con arco.



El programa de esta semana, segundo clasificado en nuestro primer concurso de programación, es una muestra de como un joven programador, Jordi Sais Riera de sólo 15 años, ha conseguido sacar muy buen partido a su ordenador.

El juego es una simulación de la clásica prueba olímpica del tiro con arco, en la cual se representan todos los factores que habitualmente dificultan su consecución, tales como la velocidad y sentido del viento, así como, por supuesto, nuestros templados nervios para disparar las flechas con la inclinación y dirección adecuadas.

TENSANDO EL COMMODORE 64 PARA OBTENER BUENOS RESULTADOS EN LAS PRUEBAS

En la pantalla inicial del juego, se nos presenta un menú de dos opciones, que nos permite escoger entre la participación de 1 ó 2 jugadores, mediante la tecla F1 y entre la entrada en competición, o bien los simples lanzamientos de entrenamiento, situación conmutable con la tecla F2. Una vez seleccionadas las condiciones deseadas, pulsaremos la barra espaciadora para comenzar el juego.

Una vez aquí, el resto del juego es realmente muy sencillo. El jugador o jugadores escriben su nombre y se preparan para la acción.

Mediante una primera acción sobre la barra espaciadora o el botón de fuego de un joystick conectado al port 1, se determinará aleatoriamente el sentido del viento, así como su velocidad (de 0 a 9), según el estado en el momento del disparo de un marcador que se altera a gran velocidad, situado en la esquina superior izquierda de la pantalla. Estas condiciones, permanecerán fijas para el resto de la partida, aunque se requerirán al comenzar un nuevo juego.

La diana se desplazará de arriba a abajo de la pantalla, debiendo nosotros pulsar la barra espaciadora (o el botón de disparo), en el momento que creamos oportuno, manteniéndola pulsada hasta que el marcador de grados situado detrás de nosotros señale la inclinación adecuada (5 grados).

Poco más queda por decir, tan sólo que seamos deportivos y que gane el mejor.







```
10 REM ** A R C O ** AUTOR: JORDI SAIS RIERA ** C.E.I
.G. ** COMMODORE 64                                  -121-
20 DIMFJ(63)                                         -166-
30 RESTORE:PRINT"{CLR}":LETPL=1:LETL$="PRACTICAR":LET
RT=4000:LETPR=1:LETGA=0:YI=0                         -105-
40 GOSUB140:GOSUB2220:GOSUB50:GOSUB430:GOSUB660:GOTO7
80                                                   -241-
50 POKE2042,194:FORI=12416TO12478:READA:POKEI,A:NEXTI
                                                     -012-
60 POKE2041,193:FORJ=12352TO12414:READB:POKEJ,B:NEXTJ
                                                     -005-
70 POKE2040,192:FORM=12288TO12350:READN:POKEM,N:NEXTM
                                                     -044-
80 POKE2046,198:FORG=12672TO12734:READH:POKEG,H:NEXTG
                                                     -030-
90 POKE2044,196:FORD=12544TO12606:READE:POKED,E:NEXTD
                                                     -008-
100 POKE2047,199:FORP=12736TO12798:READQ:POKEP,Q:NEXT
P                                                    -129-
110 POKE2043,195:FORC=12480TO12542:READF:POKEC,F:NEXT
C                                                    -044-
120 POKE2045,197:FORSW=12608TO12670:READSZ:POKESW,SZ:
NEXTSW                                               -056-
130 RETURN                                           -148-
140 POKE53280,5:POKE53281,5                          -020-
150 PRINT,"{YEL}{ABJ}{RON}[4 ESP]{DCH}[4 ESP]{DCH}[4
ESP]{DCH}[4 ESP]"                                    -203-
160 PRINT,"{RON} {2 DCH} {DCH} {2 DCH} {DCH} {4 DCH}
{2 DCH} "                                            -002-
170 PRINT,"{RON} {2 DCH} {DCH}[4 ESP]{DCH} {4 DCH} {2
 DCH} "                                              -009-
180 PRINT,"{RON}[4 ESP]{DCH} {DCH} {2 DCH} {4 DCH} {2
 DCH} "                                              -010-
190 PRINT,"{RON} {2 DCH} {DCH} {DCH} {2 DCH} {4 DCH}
{2 DCH} "                                            -005-
200 PRINT,"{RON} {2 DCH} {DCH} {DCH}[2 ESP]{DCH}[4 ES
P]{DCH}[4 ESP]"                                      -015-
210 PRINT,,"{WHT}{ABJ}BY JORDI SAIS"                 -117-
220 PRINT"{HOM}"                                     -152-
230 PRINT,"{10 ABJ}{RON}{BLU}[7 ESP]"                -109-
240 PRINT,"{RON}[2 ESP]F 1[2 ESP]";"{ROF}{BLK}[2 ESP]
";PL;"JUGADOR/ES"                                    -230-
250 PRINT,"{BLU}{RON}[7 ESP]"                        -197-
260 PRINT,"{2 ABJ}{RON}{BLU}[7 ESP]"                 -232-
270 PRINT,"{RON}[2 ESP]F 3[2 ESP]";"{ROF}{BLK}[3 ESP]
";L$;" "                                             -044-
280 PRINT,"{BLU}{RON}[7 ESP]"                        -200-
290 PRINTTAB(8)"{YEL}{ABJ}{RON}<A>!23 *!<S>"
                                                     -075-
300 PRINTTAB(8)"{RON}!-!PULSA BARRA PARA ACABAR!-!"
                                                     -187-
310 PRINTTAB(8)"{RON}<Z>!23 *!<X>"                   -161-
320 GETF$:IFF$=" "THENRETURN                         -226-
330 GETF$:IFF$=" "THENRETURN                         -227-
340 IFF$="{F1 }"THEN360                              -126-
350 GOTO380                                          -140-
360 LETPL=PL+1:IFPL=3THENPL=1                        -125-
370 GOTO220                                          -135-
380 IFF$="{F3 }"THENLK=LK+1:GOTO400                  -184-
390 GOTO330                                          -139-
400 IFLK=2THENLK=0                                   -124-
410 IFLK=0THENL$="PRACTICAR":GOTO220                 -008-
420 IFLK=1THENL$="COMPETIR":GOTO220                  -212-
430 POKE53280,14:POKE53281,5                         -070-
440 PRINT"{CLR}{BLK} <38 I>"                         -216-
450 PRINT" {RON} <A>!17 *!<S>[6 ESP]QUALIFY:[4 ESP]"
                                                     -143-
460 PRINT" {RON} !-!PUNTUACION:[6 ESP]!-![18 ESP]"
                                                     -215-
470 PRINT" {RON} <Z>!17 *!<X>[18 ESP]"               -072-
480 PRINT" {RON}[38 ESP]"                            -127-
490 PRINT" {RON} <A>!17 *!<S>[18 ESP]"               -062-
500 PRINT" {RON} !-!F. DEL VIENTO:[3 ESP]!-![18 ESP]"
                                                     -202-
510 PRINT" {RON} <Z>!17 *!<X>[18 ESP]"               -067-
520 PRINT" <38 U>"                                   -248-
530 PRINTTAB(30)"{10 ABJ}<A>!7 *!<S>"                -092-
540 PRINTTAB(30)"!-! SCORE !-!"                      -139-
550 PRINTTAB(30)"<Q>!7 *!<W>"                        -180-
560 PRINTTAB(30)"!-![7 ESP]!-!"                      -177-
570 PRINTTAB(30)"<Z>!7 *!<X>{ARB}"                   -083-
580 LETR$="_!*!"                                     -184-
590 PRINT"{HOM}{9 ABJ}{WHT}"                         -064-
600 PRINTTAB(30)"<A>!7 *!<S>"                        -176-
610 FORT=1TO5:PRINTTAB(30);"!-! ";R$;" ";R$;" !-!":NE
XTT                                                  -080-
620 PRINTTAB(30)"<Z>!7 *!<X>"                        -190-
630 PRINT"{HOM}{17 ABJ}";TAB(32);"GRADOS"            -197-
640 PRINTTAB(33);"{YEL}<[3 ESP]>"                    -204-
650 RETURN                                           -155-
660 LETV=53248:POKEV+41,7:POKEV+4,198:POKEV+5,70
                                                     -138-
670 LETV=53248:POKEV+41,7:POKEV+4,198:POKEV+5,70
                                                     -139-
680 POKEV+40,13:POKEV+3,115                          -066-
690 POKEV+37,6:POKEV+38,2                            -238-
700 POKEV+39,10:POKEV,175:POKEV+1,82                 -153-
710 POKEV+12,224:POKEV+13,185:POKEV+23,70            -137-
720 POKEV+43,2:POKEV+8,255:POKEV+9,85:POKEV+29,86
                                                     -195-
730 POKEV+14,239:POKEV+15,196:POKEV+46,13            -153-
740 POKEV+42,10:POKEV+6,255:POKEV+7,185:PQKEV+37,9:PO
KEV+38,2                                             -157-
750 POKEV+21,222                                     -145-
760 POKEV+44,10:POKEV+10,255:POKEV+11,185:POKEV+28,42
                                                     -113-
770 RETURN                                           -158-
780 GOSUB1770                                        -014-
800 IFPR=1THENPRINT"{HOM}{9 ABJ}{BLU}{5 DCH}";V$;" PU
LSA 'FUEGO'[11 ESP]":GOTO820                         -221-
810 PRINT"{HOM}{9 ABJ}{5 DCH}{BLU}";W$;" PULSA 'FUEGO
'[13 ESP]"                                           -068-
820 FORVI=1TO9                                       -138-
830 LETJV=PEEK(56321):LETFR=JVAND16:LETJV=15-(JVAND15
):IFFR<>16THEN870                                    -139-
840 PRINT"{HOM}{BLK}{6 ABJ}{BLK}"TAB(17);"{RON}[2 ESP
]";"{2 IZQ}";VI;"{4 IZQ}{RON}[2 ESP]{DCH} !-! "
                                                     -153-
850 LETAD=AD+1:IFAD=2THENAD=0                        -035-
860 NEXTVI:VU=1:GOTO820                              -252-
870 IFVU=0ANDVI=1THENVI=9                            -138-
880 IFAD=0THENVI=-VI:PRINT"{HOM}{6 ABJ}"TAB(16);"{RON
}[2 ESP]{2 IZQ}^[2 ESP]{2 IZQ}";ABS(VI):GOTO910
                                                     -028-
890 PRINT"{HOM}{6 ABJ}"TAB(17);"{RON}[2 ESP]";"{2 IZQ
}";VI;"{RON}!-!"                                     -109-
900 PRINT"{2 ABJ}[38 ESP]":IFPR=2THENRETURN          -085-
910 IFL$="PRACTICAR"THEN950                          -160-
920 FORUY=1TO3:LETPY=0:GOSUB1750                     -080-
930 LETRT=RT+1000:IFRT=8000THENRT=5000               -097-
940 PRINT"{HOM}{ABJ}{BLK}":PRINTTAB(33)"{RON}[6 ESP]"
;"{6 IZQ}";RT                                        -187-
950 GOSUB1690                                        -014-
960 FORGU=1TO11:POKE649,10:IFGA=1THENPOKEV+2,20:GOTO9
80                                                   -192-
970 GOSUB2140                                        -007-
980 B=1:Z=255:O=196:I=0:POKEV+14,239:POKEV+15,I:POKEV
+21,230                                              -063-
990 ONBGOTO1000,1150,1190                            -133-
1000 FORY=90TO255STEP2:POKEV+3,Y                     -022-
1010 IFY=>254THEN1090                                -217-
1020 IFB=1THEN1060                                   -024-
1030 IFB=4THENNEXTY                                  -237-
1040 IFB=2THEN1150                                   -027-
```

```
1050 IFB=3THEN1190                                   -033-
1060 JV=PEEK(56321):FR=JVAND16:JV=15-(JVAND15)
                                                     -248-
1070 IFFR<>16THENGOSUB2660:B=2:GOTO990               -109-
1080 NEXTY                                           -129-
1090 IFGU=10ANDOO>4THENGOSUB2750:LETGA=1:LETYI=1:LETQ
O=0:NEXTGU                                           -154-
1100 IFGU=11ANDGA=1THENGOSUB1130:LETGA=0             -253-
1110 IFGU=10THEN1500                                 -161-
1120 NEXTGU:GOTO1500                                 -248-
1130 LETLP=0:FORSW=12352TO12414:LETLP=LP+1:POKESW,FJ(
LP):NEXTSW                                           -070-
1140 POKEV+40,13:POKEV+28,42:POKEV+23,70:POKEV+29,86:
POKEV+2,0:A=0:RETURN                                 -187-
1150 IFKT=2THEN65535                                 -187-
1160 LETJV=PEEK(56321):LETFR=JVAND16:LETJV=15-(JVAND1
5)                                                   -168-
1170 IFFR<>16THENA=A+1:GOSUB2310:GOTO1190            -249-
1180 LETB=3:GOTO990                                  -150-
1190 LETZ=Z-8:IFZ<35THENGOSUB2630:LETB=4:GOSUB1220:NE
XTY                                                  -138-
1200 POKEV+14,Z:LETI=I+(.1)*VI:POKEV+15,I            -213-
1210 NEXTY                                           -124-
1220 IFGU=11ANDGA=1THENGOSUB2800:RETURN              -011-
1230 LETUI=Y+11:IFUI<ITHENMN=I-UI:AL=1:GOTO1250
                                                     -003-
1240 LETAL=2:LETMN=UI-I                              -211-
1250 IFA<5THEN1280                                   -035-
1260 LETKL=2:LETVE=A-5:GOTO1290                      -173-
1270 IFA>10THENRETURN                                -104-
1280 LETKL=1:LETVE=5-A                               -111-
1290 IFUI<YTHEN1310                                  -162-
1300 LETA=0                                          -119-
1310 LETTR=INT(MN*26+VE*52):LETA=0                   -058-
1320 LETP=750-TR:IFP<0THENP=0                        -056-
1330 IFPR=2THENGOSUB1670:GOTO1350                    -064-
1340 LETPU=PU+P:LETPY=PY+P:PRINT"{HOM}{BLK}{2 ABJ}"TA
B(14);"{RON}[6 ESP]";"{6 IZQ}";PU                    -179-
1350 PRINT"{HOM}":PRINTTAB(32)"{BLK}{21 ABJ}";P;"{IZQ
}[2 ESP]"                                            -065-
1360 IFL$="PRACTICAR"THEN1390                        -255-
1370 IFPR=2THENPY=PY+P                               -196-
1380 GOSUB1750                                       -057-
1390 IFGA=1THENRETURN                                -129-
1400 IFMN>20ORVE>10THENPOKEV+21,230:POKEV+39,1:GOSUB2
740:RETURN                                           -225-
1410 POKEV+21,103:IFKL=2THEN1430                     -127-
1420 LETAL=210-MN:GOTO1440                           -045-
1430 LETAL=210+MN                                    -240-
1440 IFKL=2THEN1460                                  -120-
1450 LETWZ=80+4*VE:GOTO1470                          -136-
1460 LETWZ=80-4*VE                                   -076-
1470 POKEV,AL:POKEV+1,WZ:FORT=1TO25:POKEV+39,1:POKEV+
39,10:NEXTT                                          -114-
1480 IFP<640THENRETURN                               -177-
1490 POKEV+21,119:FORT=1TO300:NEXTT:OO=OO+1:RETURN
                                                     -219-
1500 IFOO>5THENGOSUB2750                             -005-
1510 POKEV+21,79:LETWG=0:LETOO=0:LETTD=0             -016-
1520 IFL$="PRACTICAR"ANDPL=1THENGOSUB1660:GOSUB2690:P
OKEV+21,0:PU=0:GOTO2070                              -022-
1530 IFL$="PRACTICAR"ANDPR=1THENPR=2:GOSUB1660:GOSUB4
30:GOSUB790:GOTO910                                  -100-
1540 IFL$="PRACTICAR"THENPOKEV+21,0:GOSUB1660:GOSUB26
90:LETPU=0:PG=0:GOTO2070                             -094-
1550 IFL$="COMPETIR"ANDPL=1THEN1570                  -167-
1560 GOTO1600                                        -236-
1570 IFPY<RTTHEN2110                                 -250-
1580 LETPY=0:GOSUB1660:GOSUB590:NEXTUY               -239-
1590 GOTO1870                                        -248-
1600 IFPR=2THEN1630                                  -128-
1610 IFPY<RTTHEN2120                                 -246-
1620 LETPY=0:GOSUB1660:GOSUB590:NEXTUY:LETPR=2:GOTO91
0                                                    -039-
1630 IFPY<RTTHEN2120                                 -248-
1640 LETPY=0:GOSUB1660:GOSUB590:NEXTUY               -236-
1650 GOTO1870                                        -245-
1660 POKEV+21,95:FORT=1TO2000:NEXTT:RETURN           -201-
1670 LETPG=PG+P:PRINT"{HOM}{BLK}{ABJ}":PRINTTAB(14);"
{RON}[6 ESP]";"{6 IZQ}";PG                           -028-
1680 RETURN                                          -207-
1690 PRINT"{HOM}{9 ABJ}[39 ESP]":OO=0                -148-
1700 IFPR=2THENPOKEV+44,0:POKEV+42,0:GOTO1720
                                                     -020-
1710 PRINT"{HOM}{9 ABJ}","{BLU}";V$;" - GET READY":GO
TO1730                                               -073-
1720 PRINT"{HOM}{9 ABJ}","{BLU}";W$;" - GET READY"
                                                     -013-
1730 FORT=1TO2000:NEXTT                              -198-
1740 PRINT"{ARB}[38 ESP]":RETURN                     -040-
1750 PRINT"{HOM}{2 ABJ}{BLK}"TAB(27)"{RON}[7 ESP]";"{
7 IZQ}";PY                                           -248-
1760 RETURN                                          -206-
1770 PRINT"{HOM}{9 ABJ}"TAB(10):INPUT"JUGADOR 1";V$
                                                     -021-
1780 PRINT"{ARB}[38 ESP]"                            -018-
1790 IFLEN(V$)>8THENV$=LEFT$(V$,8)                   -004-
1800 IFPL=1THENRETURN                                -145-
1810 PRINT"{HOM}{9 ABJ}"TAB(10):INPUT"JUGADOR 2";W$
                                                     -018-
1820 IFLEN(W$)>8THENW$=LEFT$(W$,8)                   -001-
1830 PRINT"{ARB}[39 ESP]"                            -046-
1840 RETURN                                          -205-
1850 POKE2040,192:FORCO=12288TO12350:READDC:POKECO,DC
:NEXTCO                                              -212-
1860 POKEV,40:POKEV+1,YY:POKEV+39,7:POKEV+23,1:POKEV+
29,1:POKEV+21,1:RETURN                               -216-
1870 GOSUB2690                                       -065-
1880 POKE53280,2:POKE53281,2                         -074-
1890 PRINT,"{3 ABJ}"V$;":";PU;"PUNTOS"               -157-
1900 IFPU>RANDPU>PGTHENSR=R:R=PU:PRINT,"{2 ABJ}{WHT}M
UY BIEN! HAS CONSEGUIDO":GOTO1920                    -207-
1910 GOTO1950                                        -243-
1920 LETI$=V$:GOTO1930                               -053-
1930 LETYY=100:LETUU=1:GOSUB1850:PRINT,"{ABJ}NUEVO RE
CORD 'ARCO'[3 .]"                                    -226-
1940 PRINT,"{ABJ}[ANTIGUO RECORD:";SR;"]"            -067-
1950 IFPU<PGTHEN1970                                 -246-
1960 PRINT,"{ABJ}{YEL}GANADOR: ";V$                  -167-
1970 IFPL=1THEN2050                                  -128-
1980 PRINT,"{3 ABJ}{BLK}"W$;":";PG;"PUNTOS"          -032-
1990 IFPG>PUTHENPRINT,"{ABJ}{YEL}GANADOR: ";W$:GOTO20
00                                                   -024-
2000 IFPG>RANDPG>PUTHENSR=R:R=PG:PRINT,"{2 ABJ}{WHT}M
UY BIEN! HAS CONSEGUIDO":GOTO2020                    -163-
2010 GOTO2050                                        -227-
2020 LETI$=W$:GOTO2030                               -038-
2030 LETYY=150:LETF=1:GOSUB1850:PRINT,"{ABJ}NUEVO REC
ORD[3 .](ANTIGUO"                                    -039-
2040 PRINT,"{ABJ}RECORD:";SR;")"                     -117-
2050 FORT=1TO500:POKEV+39,1:POKEV+39,7:NEXTT
                                                     -001-
2060 PRINT,"{ABJ}{WHT}RECORD: ";I$;" -";R;"PTOS"
                                                     -218-
2070 PRINT"{3 ABJ}{5 DCH}{BLK}DESEA JUGAR OTRA PARTID
A (S/N)"                                             -197-
2080 GETA$:IFA$="S"THENPOKEV+21,0:LETPU=0:LETPG=0:GOT
O30                                                  -249-
2090 IFA$="N"THEN2830                                -170-
2100 GOTO2080                                        -230-
2110 LETPY=0:GOSUB2710:GOTO1870                      -166-
2120 LETPY=0:GOSUB2710:IFPR=2THEN1870                -061-
2130 LETPR=2:LETRT=4000:GOSUB430:GOSUB800:GOTO910
                                                     -082-
2140 GOSUB2310:IFWG<>0THEN2160                       -054-
2150 PRINT"{HOM}{11 ABJ}"TAB(32)"[2 ESP]":LETWG=WG+1:
RETURN                                               -166-
2160 IFWG=3THENWG=1                                  -193-
2170 IFWG=1THEN2200                                  -120-
2180 LETTD=TD+37                                     -210-
2190 PRINT"{HOM}{11 ABJ}"TAB(32+TD)"[2 ESP]";:LETWG=W
G+1:RETURN                                           -168-
2200 LETTD=TD+3                                      -148-
2210 PRINT"{HOM}{11 ABJ}"TAB(32+TD)"[2 ESP]";:LETWG=W
G+1:RETURN                                           -161-
2220 POKE53280,1:POKE53281,1:PRINT"{CLR}{BLK}"
                                                     -107-
2230 PRINT"{3 ABJ}[7 ESP]A[2 ESP]R[2 ESP]C[2 ESP]O[4
ESP](ROBIN HOOD)"                                    -069-
2240 PRINT:PRINT:PRINT"[8 ESP]AUTOR: JORDI SAIS RIERA
"                                                    -135-
2250 PRINT:PRINT:PRINT"[7 ESP]C O M M O D O R E[5 ESP
]6 4 "                                               -023-
2260 PRINT:PRINT:PRINT"[9 ESP]CONCURSO 'TU MICRO'"
                                                     -134-
2270 PRINT:PRINT:PRINT"[6 ESP]SANT FELIU DE GUIXOLS (
1985)"                                               -017-
2280 PRINT:PRINT:PRINT"{BLU}[7 ESP]PARA JUGAR UTILIZA
'FUEGO'"                                             -045-
2290 PRINT:PRINT"[5 ESP](JOYSTICK IN PORT 1) O {RON}S
PACE{ROF}"                                           -242-
2300 RETURN                                          -197-
2310 PRINT"{HOM}{18 ABJ}{YEL}";TAB(34);A;"{IZQ} > ":R
ETURN                                                -148-
2320 DATA0,255,0,3,255,192,7,255,224,15,255,240,31,25
5,248                                                -113-
2330 DATA63,255,252,127,255,254,127,255,254,127,255,2
54,127,231,254                                       -070-
2340 DATA127,231,254,127,255,254,127,255,254,127,255,
254,63,255,252                                       -071-
```





La revista semanal donde el usuario de COMMODORE podrá encontrar gran variedad de programas y noticias, sus periféricos y equipos.

SUSCRIPCION POR TELEFONO

Todos los días, excepto sábados y festivos, de **9 a 15,00 y de 17 a 19,00 horas** atenderemos sus consultas en el ☎ 2505820/2505579.
¡Suscríbase por teléfono! y en pocos días tendrá en su domicilio TU MICRO COMMODORE.

SUSCRIPCION POR CORREO

Deseo suscribirme a la revista TU MICRO COMMODORE al precio de 7.400 ptas. **(incluido IVA),** por el período de un año a partir del N.º....... inclusive.

El importe **—que abonaré en su totalidad con el primer envío—** lo haré efectivo de la siguiente forma:

☐ Talón bancario adjunto a nombre de INGELEK, S. A. ☐ Giro Postal n.º ....
☐ Tarjeta VISA n.º |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_| FIRMA
☐ Tarjeta MASTER CARD n.º |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
Fecha de caducidad de la tarjeta .... Nombre titular ....

APELLIDOS |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| NOMBRE |_|_|_|_|_|_|_|_|_|_|_|_|
DOMICILIO |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| EDAD |_|_|
CIUDAD |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| PROVINCIA |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
C POSTAL |_|_|_|_|_| TELEFONO |_|_|_|_|_|_|_|_| PROFESION |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Recorte o copie esta tarjeta
y envíela dentro del sobre a: Ediciones INGELEK.
Apartado de Correos 61 [illegible] 28080 MADRID

UERA ERRORES ¡FUERA ER

C-Byte tiene el honor de presentar, en exclusiva nacional para los lectores de TU MICRO COMMODORE, el revolucionario sistema de introducción de programas FUERA ERRORES. Este nos permitirá introducir, sin temor alguno al esfuerzo inútil, cualquier listado por largo y complicado que parezca.

Para adoptar los listados publicados bajo este sistema, deberemos seguir las siguientes normas:

1) Es fundamental transcribir EXACTAMENTE el listado reproducido, incluyendo todos sus espacios, aunque se trate de separaciones entre número de instrucción y línea de instrucción.

2) Todas las líneas finalizarán con un número de tres dígitos, encerrado entre guiones, que NO deberá ser introducido, puesto que no forma parte del programa, sino que tiene la finalidad de hacer funcionar el sistema FUERA ERRORES, según veremos más adelante. Para evitar equivocaciones, dicha cifra entre guiones se sitúa en el margen derecho del final de la línea BASIC a la cual corresponde, a una distancia prudencial del mismo.

3) Para facilitar la introducción de símbolos difícilmente interpretables, se procede a la siguiente representación en los listados.

— Las letras aparecidas entre menor y mayor deberán ser introducidas con pulsación simultánea de la tecla COMMODORE y la letra representada. Ej.: <M> =COMMODORE M.

— Las letras aparecidas entre barras verticales deberán ser introducidas como pulsación simultánea de la tecla SHIFT y la letra representada. Ej.: ¦ K¦ = SHIFT K.

— Entre corchetes simples se representarán los símbolos que se obtienen por pulsación directa de la tecla, aunque lógicamente, este caso sólo se dará para indicar las sucesiones de más de una letra. Así por ejemplo, la introducción de 5 asteriscos se representaría por [5*].

— Para la repetición de símbolos obtenidos mediante las teclas COMMODORE o SHIFT, se seguirá una combinación de las tres normas anteriormente citadas. Así por ejemplo, la introducción de 10 símbolos COMMODORE H, se representaría por [<10 H>].

— Para evitar confusiones, cuando se utilice el sistema de representación de sucesiones de carácter, y éste sea un espacio, se utilizará la abreviatura ESP. [15 ESP] = 15 espacios.

— Los carácteres de control, tales como desplazamientos del cursor, colores, estados de reversa y funciones, se simbolizarán por una abreviatura de tres letras (dos más un espacio en el caso de las funciones) encerrada entre llaves, tal como se señala en la tabla adjunta.

Para introducir cualquier listado por el sistema FUERA ERRORES, deberemos entrar previamente y ejecutar el listado BASIC que aparece en la página siguiente por lo cual es recomendable conservar una copia grabada del mismo, para sucesivas ocasiones. Una vez introducido este listado, ya sea por el teclado, o a través de cinta o disco, debemos ejecutarlo con **RUN.** Instantes más tarde aparecerá en la pantalla el mensaje **FUERA ERRORES!** y el cursor libre para la entrada de programas, con el tradicional **READY.** por encima, lo cual indicará la activación del sistema de depuración de errores.

En virtud al **NEW** que finaliza la línea 20 del programa ¡FUERA ERRORES!, este habrá desaparecido de la memoria, y seremos libres para introducir cualquiera de los programas listados en la sección **TECLA A TECLA** de cualquier número de nuestra revista, o incluso aquellos que apareciendo en otras secciones se acojan a este sistema. Así pues, si el programa ¡FUERA ERRORES! ha desaparecido de la memoria, ¿qué hemos conseguido ejecutándolo? Bien, la respuesta se llama informáticamente **INTERRUPCION:** se trata de una técnica de programación en código máquina que permite que el ordenador efectúe prácticamente dos trabajos a un tiempo, o más correctamente, que ejecute determinada tarea de forma automática, sin necesidad de que le prestemos una atención constante, de forma similar al proceso de respiración en un humano.

Efectivamente, aunque el soporte BASIC ha desaparecido de la memoria, antes de «marcharse» ha dejado funcionando en modo interrupción la pequeña rutina en código máquina que se hallaba en sus **DATAS.** Para comprobarlo pulsa **RETURN**; observarás algo muy extraño: tu ordenador no se comporta normalmente, no sólo desciende una línea el cursor, sino que además hace aparecer un número en la esquina superior izquierda de la pantalla. A continuación veremos como emplearlo.

Cada vez que pulsemos **RETURN**, aparecerá un número en la mencionada zona de la pantalla, y este corresponderá con la instrucción que hayamos introducido. Esto forma parte del sistema de FUERA ERRORES. Cuando introduzcamos cualquier línea de un listado de este tipo, deberemos fijarnos en el número que aparece al pulsar el **RETURN** de fin de línea; si este coincide con el que aprece en el listado al final de la línea, ésta habrá sido introducida correctamente, en caso contrario existe algún error de teclado que debemos modificar. Para modificar una instrucción errónea, no tenemos ni tan siquiera que volver a teclearla si no queremos, bastará sencillamente con modificar el carácter o caracteres erróneos como siempre hacemos, hasta que nos coincida el número de verificación que se presentará al pulsar **RETURN.**

Así pues, el sistema FUERA ERRORES se compone de dos partes: una codificación especial de los listados que facilitan su introducción, evitando los errores al confundir los caracteres gráficos, de control, etc... y un sistema de verificación de líneas que nos advierte en el preciso instante de introducir una de estas, que está mal tecleada.

Ahora bien, el empleo de estos dos sistemas no quiere decir que nos encontremos ante un BASIC diferente al de COMMODORE 64. Este no ha cambiado, hace exactamente las mismas cosas de siempre; simplemente hemos cambiado la forma de hacer los listados. En cuanto al misterioso número que aparece en la esquina de la pantalla no es más que

una simple suma de comprobación, lo que se conoce técnicamente como un CHECKSUM. La rutina en código máquina de interrupción suma los valores de los caracteres que entramos en cada línea que introducimos, aplicándoles un módulo 256, es decir, volviendo a cero cada vez que su suma parcial supera el 255, de este modo se crea un número entre 0 y 255 dependiente directamente de los caracteres introducidos. Así pues, es prácticamente imposible que una sucesión de errores den por casualidad ese número, mientras que siempre que la línea esté correctamente introducida obtendremos el mismo código que nosotros hallamos por igual sistema y añadimos al final de cada línea al realizar el listado del programa.

Por tanto, el misterioso número no es tampoco una modificación del BASIC de COMMODORE, sino simplemente un pequeño truco para la comprobación de que las líneas han sido bien introducidas. Propiamente no nos evita cometer errores de tecleado, sino que simplemente nos advierte inmediatamente en que línea los hemos introducido.

Para desactivar el sistema sólo deberemos pulsar RUN/STOP RESTORE, y si por cualquier motivo nos interesara reactivarlo, podríamos ejecutar SYS 822, siempre y cuando se encuentre el código máquina en la memoria, lógicamente.

¡ADVERTENCIA! Puesto que el código máquina se encuentra ubicado en el buffer del casete, es imprescindible desactivarlo (RUN/STOP RESTORE) antes de realizar cualquier operación con dicho periférico.

UTILIZACION DEL CASETE CON FUERA ERRORES

Si queremos introducir parte de un programa, para continuar posteriormente el trabajo emprendido, sin perder por supuesto la enorme ventaja del FUERA ERRORES deberemos proceder de la siguiente forma:

1. Desactivar el sistema FUERA ERRORES mediante la pulsación de RUN/STOP + RESTORE.
2. Trasladar el código máquina desde su ubicación en el buffer del casete hasta otro punto desocupado en la memoria RAM (por ejemplo 49152). Para ello sólo hemos de saber que ocupa 114 bytes desde 822 (inclusive). Una sencilla rutina que efectúe el trabajo de reubicación puede ser: FORI=0TO113:POKE49152+I,PEEK(822+I):NEXT.
3. Realizar la operación correspondiente con el casete; ya sea grabar una copia de seguridad de lo introducido en la memoria hasta el momento, ya sea cargar del casete una parte ya grabada del programa.
4. Restablecer FUERA ERRORES a su punto original. Siguiendo el caso de la línea de ejemplo anterior: FORI=0TO113:POKE822+I,PEEK(49152+I):NEXT.
5. Reactivar el sistema FUERA ERRORES mediante SYS 822 (esta vez no aparecerá mensaje alguno, aunque al pulsar **RETURN** comprobaremos que aparecen las cifras de control en la esquina superior izquierda). Lógicamente, los pasos 4 y 5 no son necesarios si después de una grabación se va a apagar el ordenador, y sólo se llevarán a cabo después de las cargas, o si tras la grabación de una copia de seguridad del programa, deseamos continuar introduciéndolo acto seguido.

TABLA DE INTERPRETACION DE CODIGOS DE CONTROL

ABR	SIFNIFICADO	OBTENCION	ABR	SIFNIFICADO	OBTENCION
HOM	HOME	CLR/HOME	F8	FUNCION 8	SHIFT F7
CLR	CLEAR + HOME	SHIFT CLR/HOME	BLK	BLACK (NEGRO)	CTRL 1
ABJ	CURSOR ABAJO	CRSR VERTICAL	WHT	WHITE (BLANCO)	CTRL 2
ARB	CURSOR ARRIBA	SHIFT CRSR	RED	RED (ROJO)	CTRL 3
		VERTICAL	CYN	CYAN (CIAN)	CTRL 4
DCH	CURSOR DERECHA	CRSR HORIZONTAL	PUR	PURPPLE (PURPURA)	CTRL 5
IZQ	CURSOR IZQUIERDA	SHIFT CSRS	GRN	GREEN (VERDE)	CTRL 6
		HORIZONTAL	BLU	BLUE (AZUL)	CTRL 7
RON	REVERSE ON	CTRL 9	YEL	YELLOW (AMARILLO)	CTRL 8
ROF	REVERSE OFF	CTRL 0	NRJ	NARANJA	COMMODORE 1
F1	FUNCION 1	F1	MRN	MARRON	COMMODORE 2
F2	FUNCION 2	SHIFT F1	RCL	ROJO CLARO	COMMODORE 3
F3	FUNCION 3	F3	GR1	GRIS 1	COMMODORE 4
F4	FUNCION 4	SHIFT F3	GR2	GRIS 2	COMMODORE 5
F5	FUNCION 5	F5	VCL	VERDE CLARO	COMMODORE 6
F6	FUNCION 6	SHIFT F5	ACL	AZUL CLARO	COMMODORE 7
F7	FUNCION 7	F7	GR3	GRIS 3	COMMODORE 8

```
10 FORI=822TO935:READA:C=C+A:POKEI,A:NEXT
20 IFC<>15254THENPRINTCHR$(147)"ATENCION!, HAY UN ERROR EN LOS DATOS":END
30 PRINTCHR$(147)TAB(213)"FUERA ERRORES!":SYS822:NEW
100 DATA 169,3,141,37,3,169,69,141,36,3,169,0,133,254,96,32,87,241,133,251
110 DATA 134,252,132,253,8,201,13,240,13,24,101,254,133,254,165,251,166,252
120 DATA 164,253,40,96,169,13,32,210,255,165,214,141,176,3,206,176,3,169,0
130 DATA 133,216,169,18,32,210,255,169,19,32,210,255,169,45,32,210,255,166
140 DATA 254,224,100,176,5,169,48,32,210,255,224,10,176,5,169,48,32,210,255
150 DATA 169,0,133,254,32,205,189,169,45,32,210,255,173,176,3,133,214,76,88,3
```

COMMODORE, COMPAÑIA AMERICANA, LIDER MUNDIAL EN NUMERO DE ORDENADORES INSTALADOS.

Su Commodore 64 tiene mucho que decirle.

Monitores.

El Commodore 64 es el resultado de la experiencia internacional de Commodore como líder indiscutible en el mercado de los microordenadores.

El Commodore 64 es el ordenador más completo y potente de su categoría,... pero todavía tiene mucho que decirle.

Por ejemplo, sus Monitores.

Monitores funcionales de diseño con pantalla de alta resolución.

commodore 64

Pensados y creados para trabajar con ellos sin notar molestias ni cansancio en la vista.

Monitores capaces de reproducir con nitidez la completa gama de colores del C-64.

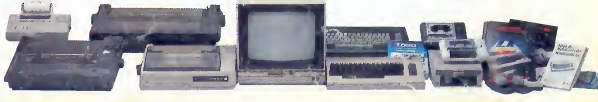
Monitores con un cuidado sistema de sonido. Con conexiones conmutables de Vídeo Compuesto y Crominancia-Luminancia.

Monitores que complementan las prestaciones de su ordenador, ganando en imagen.

Amplíe las posibilidades de su C-64, descubriendo su extensa gama de periféricos.

Ahora que ya sabe que su Commodore 64 tiene todavía mucho que decirle, prepárese a conocerle mejor.





Microelectrónica y Control, S.A. *c/ Valencia, 49-53 08015 Barcelona - c/ Ardemans, 8 28028 Madrid.*
Unico representante de Commodore en España.